1977

A Microprocessor Controlled Clock System

Joseph P. Clarke

A MICROPROCESSOR CONTROLLED CLOCK SYSTEM

BY

JOSEPH P. CLARKE

A thesis submitted
in partial fulfillment of the requirements for the
degree Master of Science, Major in
Engineering, South Dakota
State University
1977

A MICROPROCESSOR CONTROLLED

CLOCK SYSTEM

This thesis is approved as a creditable and independent investigation by a candidate for the degree, Master of Science, and is acceptable as meeting the thesis requirements for this degree. Acceptance of this thesis does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

ACKNOWLEDGMENTS

The author expresses sincere appreciation to Dr. D. E. Sander and the many other persons who gave their advice, suggestions, and encouragement.

J. P. C.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

An Introduction to Timekeeping

In the story by J. R. R. Toelkin, The Hobbit, Bilbo Biggins was given a riddle to decide his very existence.

This thing all things devours:
Birds, beasts, trees, flowers;
Gnaws iron, bites steel;
Grinds hard stones to meal;
Slays king, ruins town;
And beats high mountain down.

Because of the gravity of the situation, Bilbo needed a considerable period to ponder the riddle. So Bilbo squealed:

"Time! Time!"

Which by pure luck was the answer.

Time and its measurement has interested man since the beginning of time. The earliest time measuring devices were sundials and water clocks. The clepsydra, a water clock linked to a gearing system was the object of a considerable amount of man's ingenuity during the first millennium A.D. A notable example was a clepsydra given to Charlemagne by the King of Persia in 807.[2] This time piece not only had a dial but rang the hour.

In the thirteenth century mechanical escapement clocks began to appear. Time was kept on these early clocks by a foliot or cross bar which oscillated in a horizontal plane. In 1656 Christian Huygens adopted the vertical pendulum of Galileo which is still used in many long case and wall clocks today.[3] The introduction of the mainspring and balance wheel, along with the myriad materials and

compensation improvements, have created highly accurate and reliable time pieces.

Clocks with complex gearing systems indicating more than just time have been created since the fourteenth century. In 1364 Giovanni De'Dondi completed an astronomical clock that gave the positions of the sun, moon, and planets. This clock also displayed calendar and zodialogical information along with the dates of fluctuating religious days. Leonardo da Vinci, in the decade after 1489, made many drawings of the gearing systems of De'Dondi's clock. These drawings still survive.[4]

Another type of clock which was developed during this time was the hour strike and alarm clock. The alarm clock first appeared at the end of the fourteenth century and had a pin placed in the "hour" wheel-gear to select the alarm time. This type of clock achieved wide acceptance in monasteries throughout Europe to announce religious and monastic functions throughout the day and night.

Mechanical decision making logic, like that in the early alarm clocks, changed little in the next few centuries while man's ingenuity was devoted to making time pieces more accurate, reliable and attractive. In contrast to the astronomers and clock makers, the mathematicians began to show interest in mechanical logic machines.

While the abacus dates before 450 B.C.,[5] little work was done in developing mechanical calculators until Blaise Pascal designed in 1642 a mechanical computer that performed addition and subtraction.[6] This device was not a computer in the modern sense, but

rather a geared adding machine. It took another eighty years and the genius of Charles Babbage to develop the first computer.

In 1822 Babbage completed his "Difference Engine" that was used in calculation of polynomials.[7] With the success of this machine he proposed a "Difference Engine" accurate to twenty places rather than the six places of the earlier machine. Due to the lack of precise machine shop equipment the second "Difference Engine" was never completed. In 1834 Babbage proposed an "Analytical Engine." This machine was a programmable computer that contained a central processing unit or "mill." It also contained a memory system called a "store" which would have had the capability of fifty digit accuracy. Due to technical limitations of the time, the project was never attempted. Many refinements in mechanical computational devices followed, but it was not until the 1930's that serious work on producing an electrical computer was begun. In 1943 the Harvard Mark I, an electromechanical machine was completed.[8] To increase speed a purely electronic machine using vacuum tubes called ENIAC was put into use in 1946. The stored program computer envisioned by Dr. John von Neumann was completed in 1952.[9] Since that time there has been a constant increase in the size and speed of computers. The advances in solid state technology have caused a rapid evolution from vacuum tubes to transistors to integrated circuits.

The introduction of large scale integrated circuitry made it possible to incorporate an entire calculator on four to eight chips

and later one chip. Busicom, a Japanese manufacturing company, contracted Intel to produce integrated circuit chips for a programmable calculator. The original eleven chip design was reduced to three by Ted Hoff, who had previously worked on the classical 1103 memory chip.[10] The three chips were known in 1971 as the 4004 central processing unit and two memory chips. At about the same time Intel, in conjunction with Datapoint and Viatron, developed a very simple eight bit computer on a chip.[11] The chip that was developed was only one tenth as fast as that needed by Datapoint so Datapoint implemented their product in random logic. Viatron had encountered serious financial and management problems and declared bankruptcy. Intel, having lost its two major customers, marketed the eight bit logic unit as the 8008 microprocessor in 1972.[12] In the following four years every major semiconductor manufacturer produced a competitive family of microprocessors. The microprocessor has truly become a cornerstone in electronic design.

While the areas of time keeping and computers have been somewhat independent, there have been exceptions. The commercial time recording clocks have for many years been used to ring bells in schools and factories. An example of this type of system is the simplex control clock designed by International Business Machines in 1949.[13] This type of clock relies on electromechanical logic. With the introduction of low cost digital logic and microprocessors, the designer has been given the opportunity to develop realtime control systems and sophisticated digital clocks.

The first attempt at this project definition was to develop a digital clock and some type of electronic storage and decision making system that would perform the functions of the Simplex system. Many approaches were considered and rejected before the complete system was defined. It was finally decided to have a digital clock connected to a microcomputer system that would compare the time to a table of times and commands in a table in read only memory. The computer would send commands to a series of circuits that would finally perform a variety of tasks. Features included which perform additional tasks to the simplex system include a calendar, intrusion alarm, westminster chime system and a programmable cycle indicator. An important aspect of this project was the area of modularity and system layout. Several years ago the author was fortunate to have worked on the construction of a bowling score calculator to assist blind bowlers. This was a master's project designed by Gary Wilhelm.[14] Problems were encountered in construction that were not unlike those of Charles Babbage and the "Analytical Engine."[15] The genius of design was too technically intensive for the materials at the time. Significant improvements have been made since that time in the area of prototype components. Now it would be much easier to implement the bowling computer by utilizing modular construction. The experience gained from the bowling computer illustrated that successful completion of a complex clock would require breaking up functions into modules.

By distributing the system on a number of small boards each

having an independent task, errors in design and construction can by located easily and corrected. Except for the power supply individual boards can be checked without other boards placed in the system. Sequential circuitry is avoided wherever possible to speed system diagnostics and improve noise immunity. Another advantage of modular systems is the flexibility of the finished product. With a minimal set of modules a system can be configured in many ways by using different combinations of modules. Another aspect of flexibility is the opportunity of system expansion if the need arises.

To demonstrate this modular concept, a number of responding output boards were created. These boards incorporate a variety of functions, design concepts, and device technologies to demonstrate the versatility of the system.

Another factor that affected the overall design of the clock was the fact that it was to be primarily a demonstration prototype. This had the effect of making the design somewhat conservative at the expense of cost of optimization. These factors also added a number of controls and indicators that would not be needed in production models. The designs were achieved using over 150 integrated circuits, 500 miscellaneous components, 4100 interconnections, and two year's work.

The detailed discussion covering the clock system and the individual modules that follows will assume a knowledge of electronics and digital logic on the part of the reader. If tutorial information

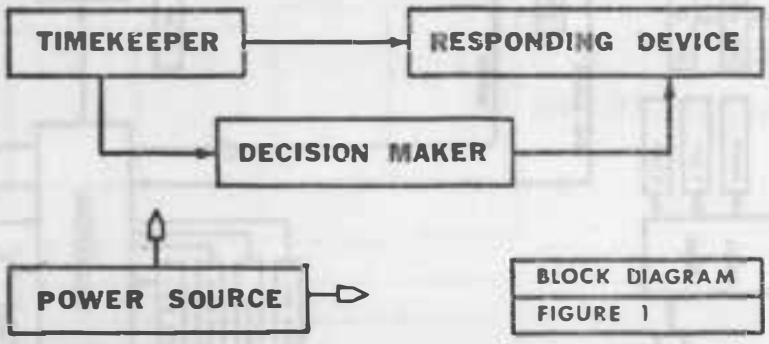
is needed, the works of Lancaster and Wicks listed in the references should be consulted. Conventions maintained to improve schematic legibility will be the omission of power and ground lines, decoupling capaciters, and data code tables from the schematics. The data code tables are listed in the appendix for reference. Because of the vast nature of this project, it would be impossible to be both complete and concise enough to satisfy all readers. Much of the background information, component specifications, and software documentation is referenced in the footnotes.

CHAPTER II

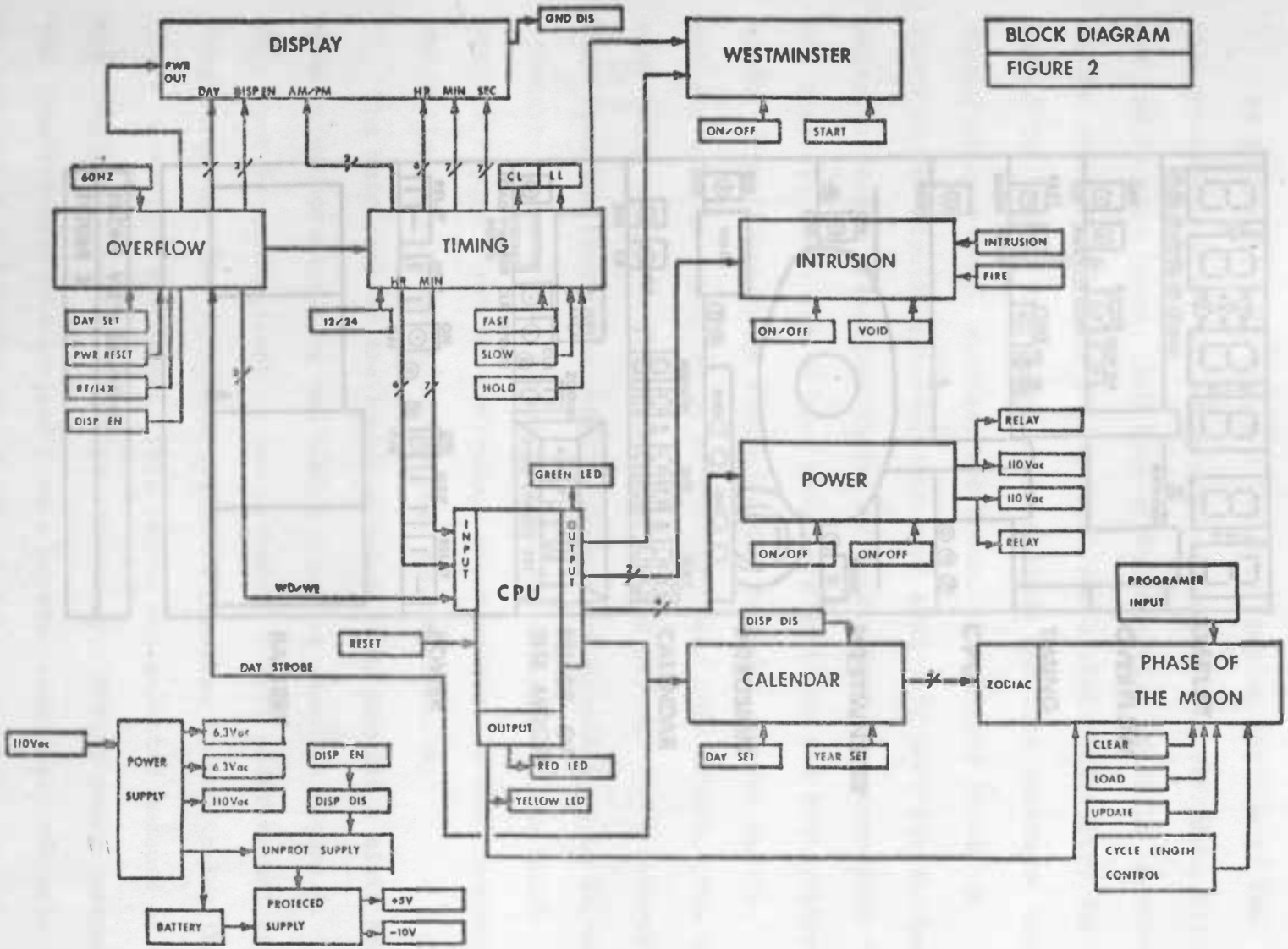
Timing and Decision Sections of the Project

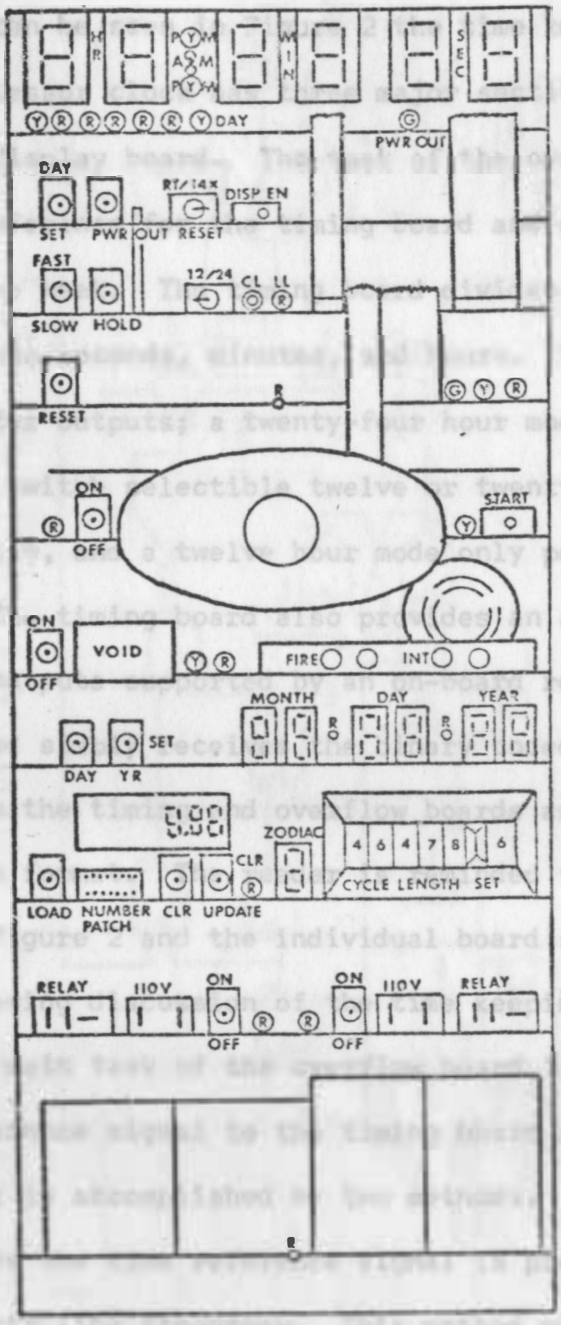
A. Introduction

Programmable time pieces from alarm clocks to sophisticated real time control systems can be analyzed in four parts; the time keeper, the decision maker, the responding device, and the power source, as shown in Figure 1. In a simple wind up alarm clock the time keeper contains a spring and balance wheel that oscillates at uniform frequency, and an escapement and gear system that yields the minutes and hours. The decision maker is a cam action device that decides if the time is equal to the alarm setting. The responding device in an alarm clock is the bell, and the power source is the main spring. While the previous discussion may seem overly elementary, it is well to keep this simple model in mind when examining the microprocessor clock system as shown in Figures 2 and 3.



BLOCK DIAGRAM
FIGURE 1


DISPLAY
GND DIS
PWR OUT
DAY
DISP EN
AM/PM
HR
MIN
SEC
WESTMINSTER
ON/OFF
START
BLOCK DIAGRAM
FIGURE 2
60HZ
CL
LL
OVERFLOW
TIMING
HR
MIN
INTRUSION
INTRUSION
FIRE
ON/OFF
VOID
DAY SET
PWR RESET
12/24
DISP EN
12/24
FAST
SLOW
HOLD
RELAY
110Vac
110Vac
RELAY
POWER
ON/OFF
ON/OFF
GREEN LED
INPUT
CPU
OUTPUT
WD/WE
PROGRAMER INPUT
DISP DIS
RESET
DAY STROBE
CALENDAR
ZODIAC
PHASE OF THE MOON
DAY SET
YEAR SET
CLEAR
LOAD
UPDATE
CYCLE LENGTH CONTROL
OUTPUT
RED LED
YELLOW LED
110Vac
POWER SUPPLY
6.3Vac
6.3Vac
110Vac
DISP EN
DISP DIS
UNPROT SUPPLY
BATTERY
PROTECED SUPPLY
+5V
-10V

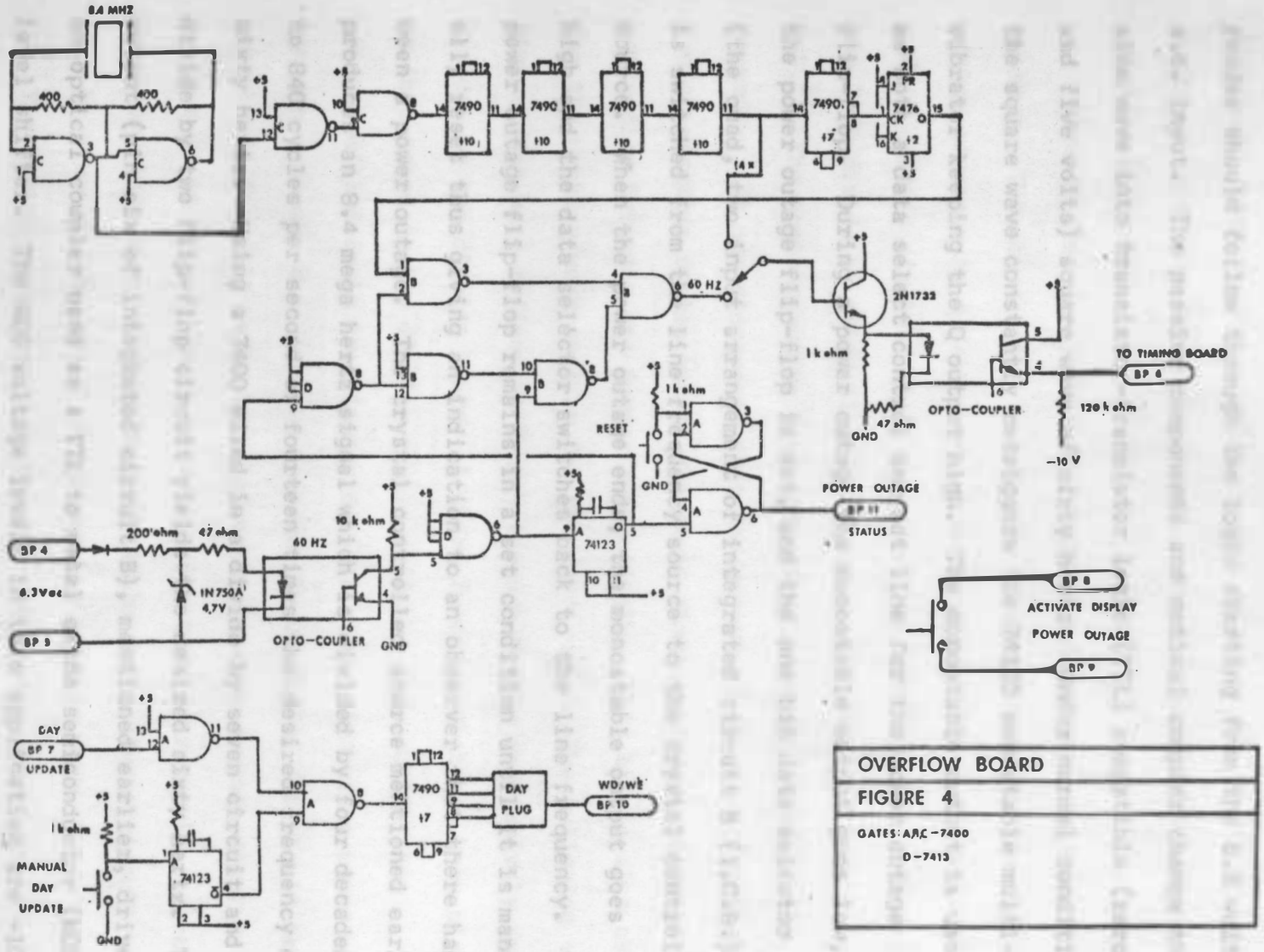


FRONT VIEW DRAWING
FIGURE 3

B. Timing Section

As can be seen in Figure 2 the time keeping section of the microprocessor clock has three major sections; the overflow, timing, and the display board. The task of the overflow board is to provide a time reference for the timing board and division circuitry for the day of the week. The timing board divides the time reference frequency into seconds, minutes, and hours. The timing board has three major outputs; a twenty-four hour mode only port for the computer, a switch selectible twelve or twenty-four hour mode port for the display, and a twelve hour mode only port for the Westminster board. The timing board also provides an am/pm and two special purpose outputs supported by an on-board read only memory. The display board simply receives the binary coded decimal (BCD) information from the timing and overflow boards and displays it in an understandable format. The reader is reminded to refer to the block diagram Figure 2 and the individual board schematics when reading the following discussion of the time keeping section.

The main task of the overflow board is to provide a sixty hertz time reference signal to the timing board as shown in Figure 4. This task is accomplished by two methods. First, under normal circumstances the time reference signal is produced by monitoring the sixty hertz line frequency. This method was selected because of the long term accuracy of utility frequency. During power outages the time reference is derived from a crystal oscillator circuit. To understand exactly how the frequency reference functions, the



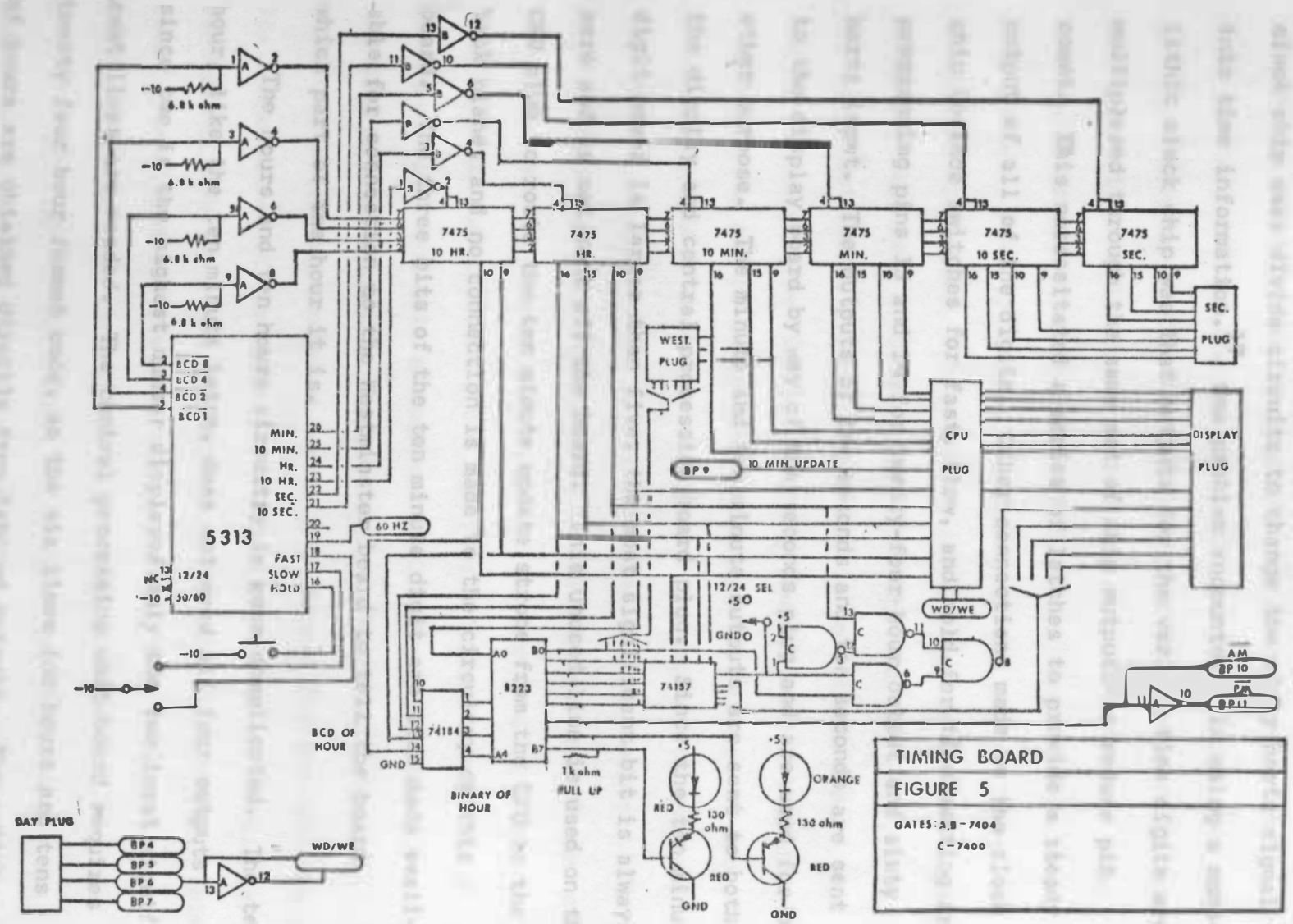
OVERFLOW BOARD
FIGURE 4
GATES: A,B,C – 7400 D – 7413

reader should follow through the logic starting from the 6.3 volt a.c. input. The passive components and optical coupler change the sine wave into transistor-transistor logic (TTL) compatible (zero and five volts) square wave of sixty hertz. Under normal conditions the square wave constantly retriggers the 74123 monostable multivibrator keeping the Q output high. The monostable output is used as both a data select control and set line for the power outage flip-flop. During a power outage the monostable output goes low, the power outage flip-flop is set, and the one bit data selector (the quad, two input arrangement of integrated circuit B (I.C.B.)) is switched from the line frequency source to the crystal controlled source. When the power outage ends, the monostable output goes high and the data selector switches back to the line frequency. The power outage flip-flop remains in a set condition until it is manually reset thus giving an indication to an observer that there has been a power outage. The crystal controlled source mentioned earlier produces an 8.4 mega hertz signal which is divided by four decades to 840 cycles per second or fourteen times the desired frequency of sixty hertz. Using a 7400 wired in a divide by seven circuit and a divide by two flip-flop circuit yields the desired sixty hertz. The output (pin six of integrated circuit B), mentioned earlier, drives an optical coupler used as a TTL to metal oxide semiconductor (MOS) level shifter. The MOS voltage levels in this application are -10 and +5 volts. As can be seen by the schematic, the level shifter can also be switched from the sixty hertz data selector to the

840 hertz point in the crystal circuit to drive the time keeping system at fourteen times the normal speed for diagnostic and demonstration purposes.

Another function performed by the overflow board is the day counter. The day counter is a divide by seven counter than can be either advanced by a debounced update switch or the computer day update command strobe. It is important to note that the output of the 7490 divide by seven circuit is not a uniform count 0-6 or 1-7.[16] The nonconsecutive four bit output code is listed in the appendix. It was discovered that the output of Qc was a logic one, two consecutive states out of the seven states. These states were defined as Saturday and Sunday respectively. This bit is also used for the weekday/weekend flag for the day's code. The following five states were defined as Monday through Friday respectively and decoded by the display board. A third function that has been placed on this board is the display enable switch. This switch when pressed will default the display disable circuitry during power outages and make the displays momentarily readable. It should be remembered that all major logic circuits (except certain displays and output circuits) have a battery protected power supply. A more detailed discussion of the display disable and activate circuitry will be given in the power source section of this work.

The timing board receives the time reference signal from the overflow board and uses a National semiconductor 5313 clock chip to generate the seconds, minutes, and hours as shown in Figure 5. The



TIMING BOARD

FIGURE 5

GATES: A,B - 7404
C - 7400

clock chip uses divide circuits to change the sixty hertz signal into time information.[17] One problem encountered in using a monolithic clock chip was that outputs for the various time digits are multiplexed through the same set of chip outputs to reduce pin count. This necessitated a series of latches to provide a steady output of all of the digits. Other connections made to the clock chip include switches for fast, slow, and hold for time setting and programming pins 13 and 14 for twenty-four hour output and sixty hertz input. The outputs of the seconds and ten seconds are sent to the display board by way of the seconds plug and are used for no other purpose. The minute and ten minute outputs are sent to both the display and central processing board plug. Since the ten minute digit never is larger than five, the most significant bit is always zero and is not sent off the board. This unused line is used on the CPU plug to route the ten minute update strobe from the CPU to the back plane, and no connection is made to the circuitry on this board. The three bits of the ten minute digit are also made available for connection to the Westminster board to tell the board which part of the hour it is.

The hours and ten hours circuitry is more complicated. The ten hour, like the ten minute latch, does not need all four outputs since two is the highest number displayed only the two least significant lines are needed. The central processing unit board requires twenty four hour format code, so the six lines for hours and tens of hours are obtained directly from latched outputs. The switch

selectable twelve or twenty-four hour format requires a novel circuit approach. First the two digit BCD code is changed into the binary equivalent of the two digits using a 74184 read only memory code converter. This binary code is fed into an 8223 read only memory, programmed to output the BCD code for the hours in twelve hour format. Five of the eight output bits are used to generate the twelve hour code. The remaining three bits are used for: one, an am/pm flag; two, a coffee break light that is on at ten, two, and four; and three, a lunch light that is on between twelve noon and one o'clock. The lights, amusingly programmed for this demonstration unit, can be programmed to indicate any number of specific hours that would be desired. The twelve hour code from the read only memory is supplied to a five bit data selector network. The data selector is also supplied with the twenty four hour format code.

The five bit data selector network is comprised of a 74157 four bit data selector and a one bit data selector made out of a quad two input nand gate arrangement (I.C.C. in Figure 5). The five bit data select network is controlled by a switch which has the effect of selecting twelve or twenty four format for the display. The output lines of the data selector are connected to the display plug along with the least significant bit of the hours digit. This least significant bit does not change with an hour format change and need not be sent through the data select network.

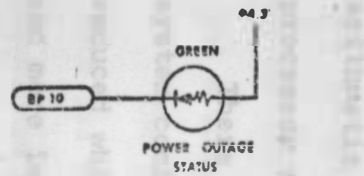
Another type of hour output was created for the possible use with hour strike circuitry where twelve hour only format would be used.

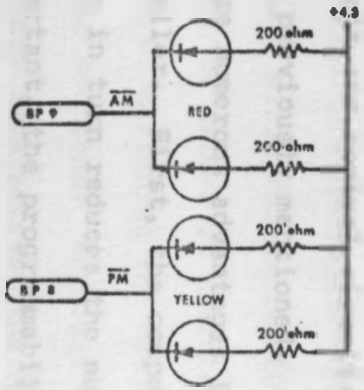
This output was obtained by connecting lines to the twelve hour code generator ROM and to the Westminster plug. The timing board thus has three formats of hour output; twenty four hour only, twelve hour only and selectible.

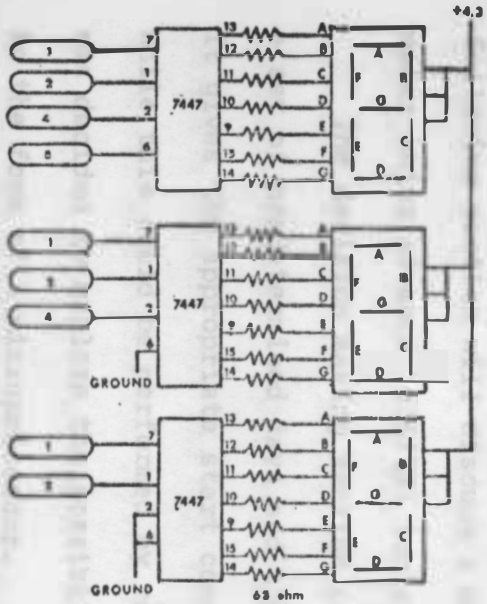
The day code bus also passes through this board from the day plug to the back plane. The weekend flag is inverted to give both the weekend and weekday flag to the CPU plug for use by the microprocessor. A complete list of plug and cable codes can be found in the appendix.

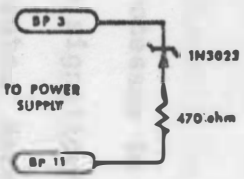
The display board receives the binary coded decimal information of the time, the flag lines of power outage, am and pm, and also receives the day code as shown in Figure 6. The display board decodes this information and displays it in a readable format. The time display is a standard common anode arrangement using 7447 decoder driver chip. Where certain logic lines are uniformally low, as in the case of the tens of seconds, minutes, and hours, they have been tied low. The a.m. (red) colon, p.m. (yellow) colon, and power outage light are active low. The day display makes use of a 74154 four to sixteen line decoder where a certain output line goes low for each input code on the four input lines. The day code for Sunday to Saturday is 5, 9, 0, 1, 2, 3, 4. By connecting the row of day lights to the proper outputs on the four to sixteen decoder, the day of the week can be properly displayed.

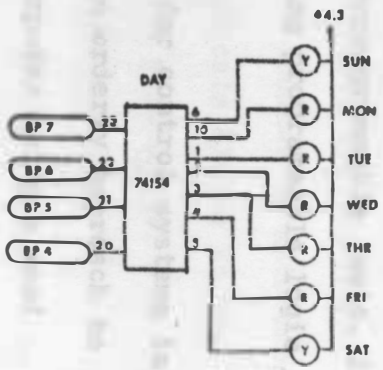
The time keeping section just discussed fulfills the same task that the spring and balance wheel and gears dial do in a simple











DISPLAY BOARD

FIGURE 6

alarm clock.

C. Decision Section

The decision making section as in an alarm clock, has to answer the question: Is there anything to be done at this time? The following section will discuss a method of implementing a decision maker using advanced digital logic.

The decision making section compares the actual time and day to a set of predetermined days and times. If the day and times match it gives the appropriate start command to a predetermined section. While this could be performed by random or nonprogrammable logic, it was decided to explore the possibility of using programmable logic in the form of a microprocessor.

The concept of using programmable logic for control systems is not new. M. V. Wilkes in 1951 conceived of an orderly approach to designing the control section of a digital computer using a read only memory and arithmetic unit.[18] With the advent of large scale integrated circuitry, it was possible to incorporate the entire arithmetic unit and control circuitry into a single chip or microprocessor as previously mentioned.

There are numerous advantages in using a microprocessor in a system controller. First, the component count can be significantly reduced which in turn reduces the number of interconnects. Second, and more important, the programmability reduces the redesign costs for both correction of original design errors and performance modifications needed by new product applications. In one sense, the

clock system required a certain level of programmability for the specific alarm times. The reasons mentioned, along with the many subtle advantages associated with microprocessors, made the microprocessor based system controller the only logical choice. Microprocessors, of course, are not without their problems. For many designers not acquainted with computer software, the problems of creating the process algorithms and translating them into machine language commands, can at first seem insurmountable. It is the author's opinion that the only effective way to become acquainted with microprocessors and their software is to study systems and programs already in existence. There are many sources of materials on existing microcomputer systems. The periodicals listed in the appendix, microprocessor vendors, and companies merchandising personal computing systems have a wealth of valuable information.

The specific control system for the clock system was developed after studying four different microcomputers for a period of a year. The decision making system that evolved was influenced greatly by the Martin Research Microcomputer Design Book.[19] This book covers primarily Intel microprocessors, but should be considered a necessity and a fundamental primer on microcomputer hardware.

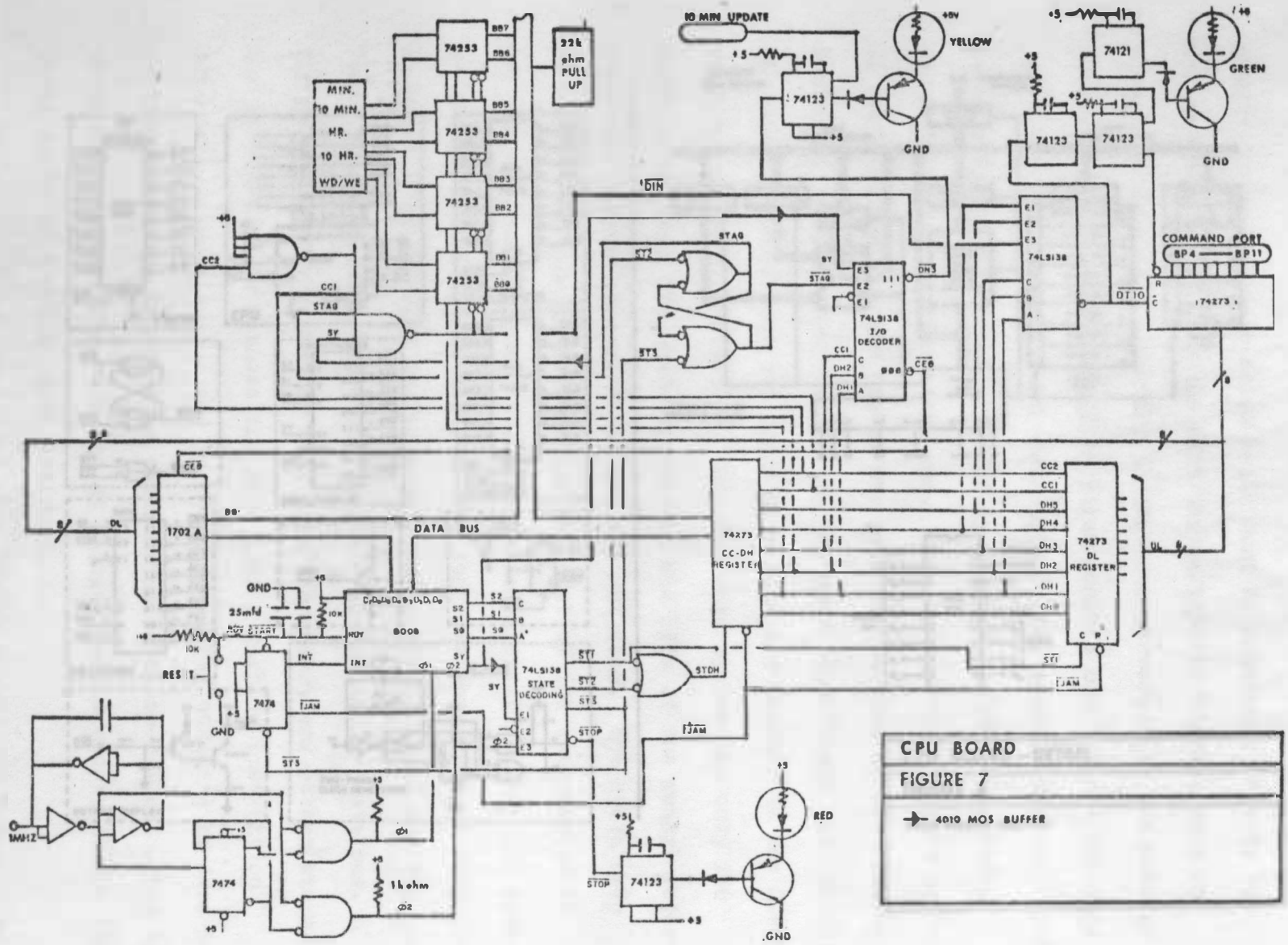
The task of designing the decision making section can be divided into two major sections: hardware and software. While the two sections are highly interrelated the bulk of the software programming will be discussed after the hardware has been covered. The hardware can be further divided into two sections: the main computer and

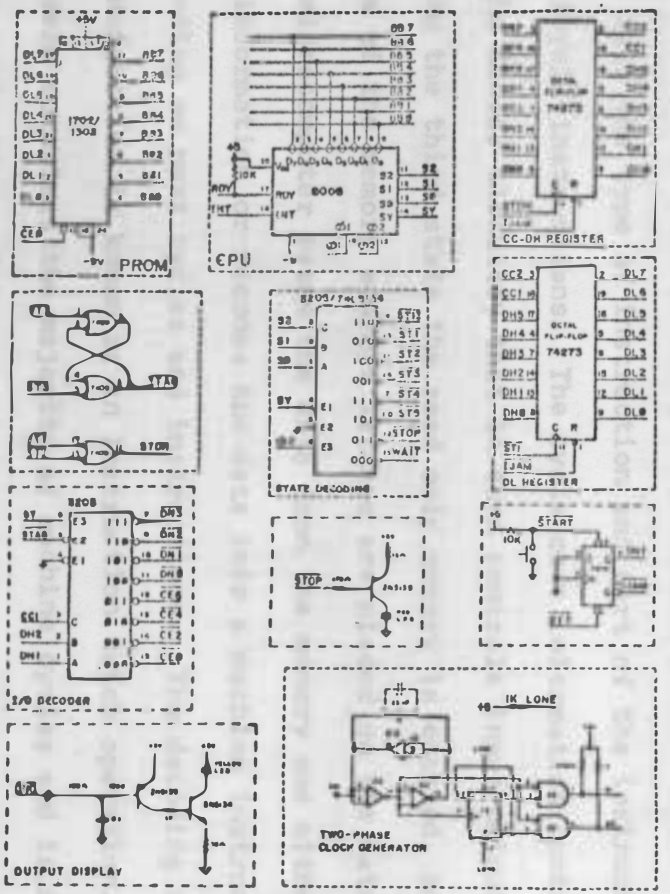
the input-output circuitry. Figures 7 and 8 illustrate the input-output circuitry (top half of drawing) and the main computer (bottom half of drawing).

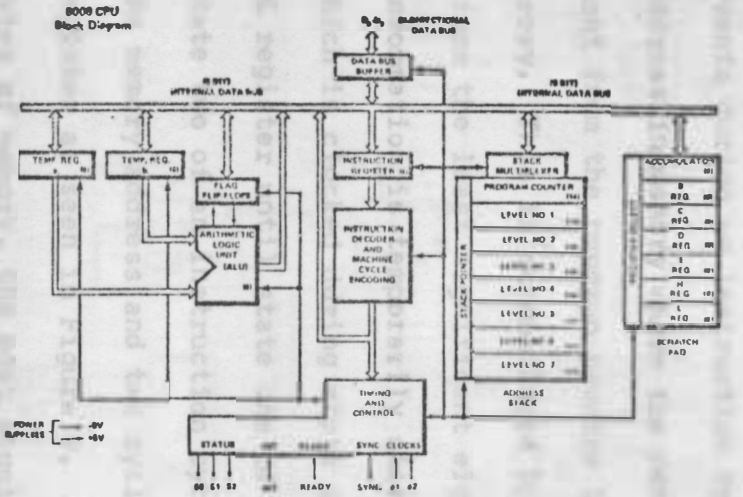
The main computer is noteworthy because of its minimal configuration of only two MOS integrated circuits and seven TTL integrated circuits. Another interesting feature of this design is the absence of any random access (RAM) memory. The program requirements for random access memory are handled by the microprocessor's registers. This has the effect of eliminating not only the RAM chips but also some decoding circuitry.

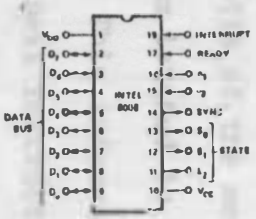
The main computer, although small in component count, would require several hundred pages documentation to fully describe all timing diagrams, circuit parameters, and the intricacies of the software. No attempt will be made to repeat background material presented in the Intel and Martin research publications mentioned. The discussion that follows is a brief overview of the main computer and its functions.

The heart of the microcomputer is an Intel 8008 eight bit microprocessor. The microprocessor performs arithmetic and logical operations, controls the data flow in the system, and controls the input-output circuitry. Because of the pin number limitation at the time of development of the 8008, the eight data pins are used for multiple functions in a time multiplexed format.[20] To retain this multiplexed data external latches controlled by state decoding circuitry were added. To illustrate how this latch network works, the sequence of


10 MIN UPDATE
+5V
YELLOW
GND
74123
74121
GREEN
74123
74123
74253
74253
74253
74253
MIN.
10 MIN.
HR.
10 HR.
WD/WE
22k ohm PULL UP
DIN
STAG
ST2
ST3
SY
74LS138 I/O DECODER
74LS138
COMMAND PORT
BP4 — BP11
74273
DT10
CC2
CC1
STAG
SY
CE0
1702 A
DL
DATA BUS
8008
74273 CC-DH REGISTER
74273 DL REGISTER
DH5
DH4
DH3
DH2
DH1
DH0
GND
25mfd
10K
RESET
INT
7474
IJAM
74LS138 STATE DECODING
STDH
STOP
1MHZ
1k ohm
Φ1
Φ2
RED
CPU BOARD
FIGURE 7
4010 MOS BUFFER







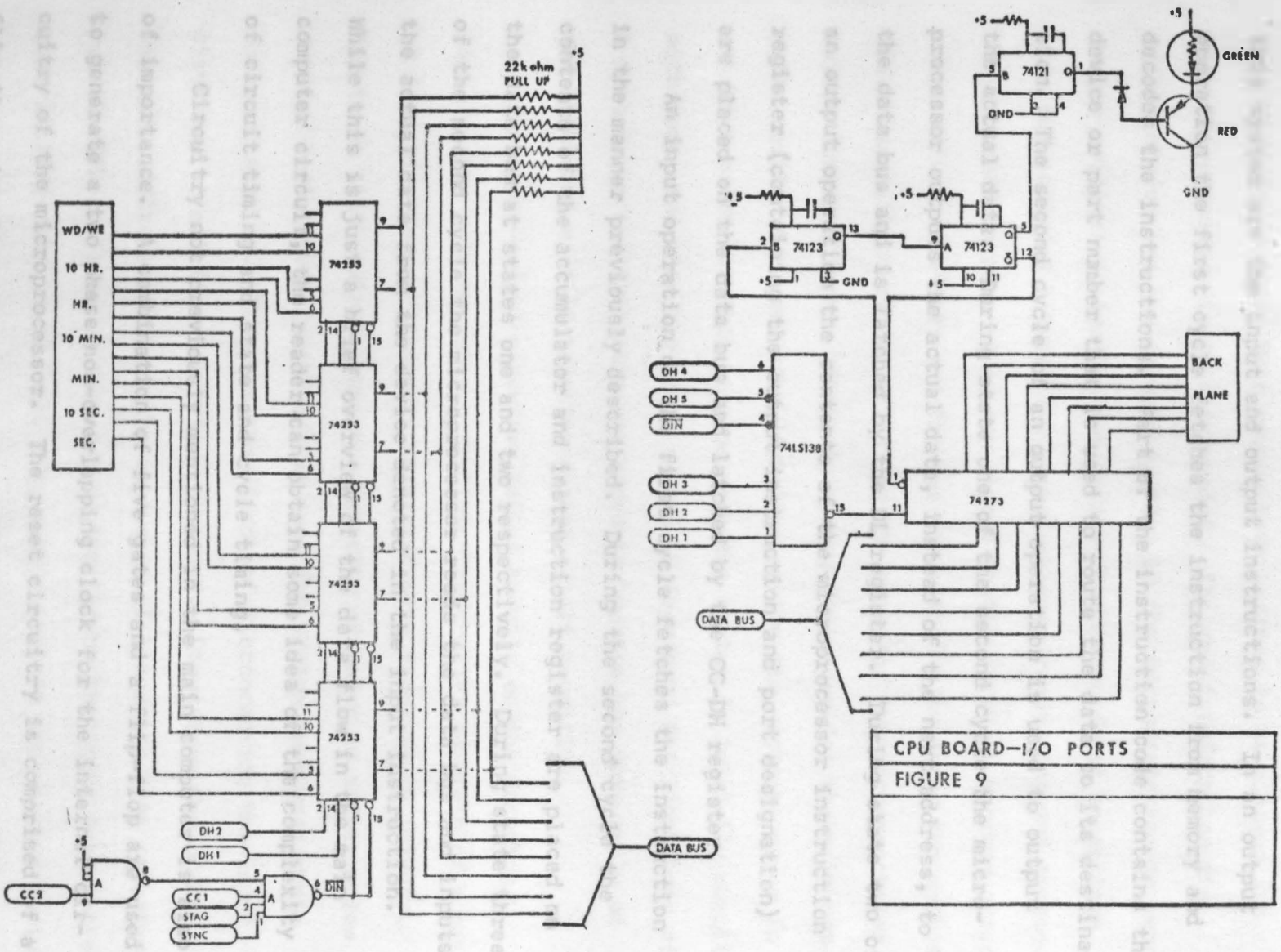
CPU BOARD—DETAIL

FIGURE 8

FROM: MARTIN RESEARCH

events during an instruction cycle must be observed. First, the address in memory where the next instruction is stored, must be sent from the program counter in the microprocessor to the memory array. This is accomplished by first sending out through the data lines the least significant eight bits of the memory address. This information is temporarily stored or latched in the DL register which is clocked during state one. This information remains in the DL register until state one of the next instruction cycle. During state two of an instruction cycle, the most significant six bits of the memory address and two cycle code bits are latched in the CC-DH register as seen in Figure 9. Because this system has only 256 bytes of memory, the most significant six bits of the memory reference address are always zero, they are not used. The cycle code bits tell the type of instruction and part of the instruction in multicycle instructions. The cycle code information controls some of the decode circuitry that in turn controls input-output circuitry. During the third state the read only memory is enabled, and the contents of the memory address latches are placed on the data bus. The microcomputer reads the data from the memory and either stores the information or decodes the data into a machine instruction depending on past cycles and instructions. The decoding of data into an instruction is known as an instruction fetch operation. This type of cycle comprises the majority of machine cycles and is always the first cycle of a multi cycle instruction.

Two multi cycle operations that are of particular interest in



CPU BOARD—I/O PORTS

FIGURE 9

this system are the input and output instructions. In an output operation the first cycle fetches the instruction from memory and decodes the instructions. Part of the instruction code contains the device or part number that is used to route the data to its destination. The second cycle of an output operation is used to output the actual data. During state one of the second cycle the microprocessor outputs the actual data, instead of the next address, to the data bus and is latched by the DL register. During state two of an output operation the contents of the microprocessor instruction register (containing the output instruction and port designation) are placed on the data bus and latched by the CC-DH register.

An input operation on the first cycle fetches the instruction in the manner previously described. During the second cycle the contents of the accumulator and instruction register are placed on the data bus at states one and two respectively. During state three of the second cycle the microprocessor reads the data bus and inputs the actual data from the device denoted in the input instruction. While this is just a brief overview of the data flow in the main computer circuit, the reader can obtain some idea of the complexity of circuit timing and state and cycle timing.

Circuitry not previously mentioned in the main computer is also of importance. A combination of five gates and a flip-flop are used to generate a two phase non-overlapping clock for the internal circuitry of the microprocessor. The reset circuitry is comprised of a flip-flop circuit that when the reset switch is depressed the

flip-flop is preset. This causes both latches to be cleared and the microprocessor to be interrupted. The reset switch has the effect of forcing the program counter in the microprocessor to zero. In the original design the two twenty-five micro-farad capacitors were not included between the microprocessor ready line and ground.[21] It was found because of switch bounce that a delay in microprocessor response was necessary and the two capacitors eliminated the problem. These capacitors have no effect on the speed of the microprocessor except during a reset. The only system indicator for the main computer section is a red stop light indicating the microprocessor in a halt condition. This happens when the system is initially supplied with power and when the reset switch is depressed the light goes off. A monostable multivibrator was added to improve the duty cycle of the light and make it brighter during a halt condition.

The input-output circuitry relies heavily on the decoding logic in the main computer. The input controller consists of a set of four line to one line data selectors with three state outputs. The three state outputs have the capacity of being in either a low or high logic level in the active mode and a high impedance state when not enabled. This has the effect of disconnecting the input circuitry at all times except when needed. Lines from the CC-DH register controls which of the four sets of lines the data selectors select. These lines in turn are controlled by the input instruction that includes the input port. The data selectors are enabled during state three of the second cycle of both input and output operations.

In output operations the microprocessor simply ignores the data in the data bus, and the input circuitry has no effect on the general system. Only two of the four input ports are presently used, making the system easily expandable. The ports have been wired so that they are port zero for days and hours, port one for minutes, and ports three and four not used. Because an 8008 input requires a logic one to be at least 3.5 volts and many TTL logic outputs are not guaranteed to be over 2.4 volts in a logic one state, a set of eight 22k ohm pull up resistors were added, one on each data line, to have a sufficiently high voltage for a logic one voltage.

There are two types of output ports in the output controlier: a standard eight bit port referenced by an output command and a single bit port actually referenced by an input instruction. For the eight bit output port the decoding circuitry was wired to yield an enable strobe for eight bit latch when the CC-DH register indicated output port ten. Two monostables to the port enable the strobe to give a delayed clear to the output latch. The output code remains for about 100 micro-seconds, and then resets to zero. This gives the effect of giving the appropriate line a short positive pulse, which the response boards are designed to accept. Another monostable and a green LED indicator were wired to the reset circuit to indicate that the computer has issued an output on this port. This indicator is used for confirming circuit performance and diagnostic work.

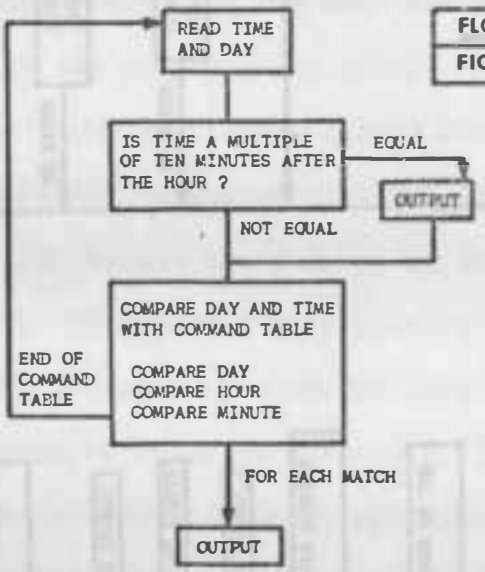
The single bit output port is actually an enable strobe for input port three. Used properly in the program software this input

port strobe can provide any outside system a one bit start strobe. The only additional circuitry that was added to complete this port was one monostable and a yellow LED indicator for diagnostic purposes. The input command in this case fills the accumulator with ones, but since the accumulator is not used in this operation and is written over following this operation, there is no disadvantages in using this novel hardware approach.

The designer that has control over both hardware and software development has a great deal of freedom in system development. The designer that is well acquainted with digital logic and new to microprocessors and software will implement most of the design task in random logic and only the central decision making circuitry will involve the microprocessor. This cautious approach is quite justifiable in the beginning because there is a great deal to learn about even in a minimal microcomputer system. In following projects the designer will fulfill more and more tasks using software and processor support circuits rather than random logic. The clock system under discussion is an example of a first generation microprocessor system with only the central decision making system a microprocessor oriented system. The program while somewhat simple warrants discussion.

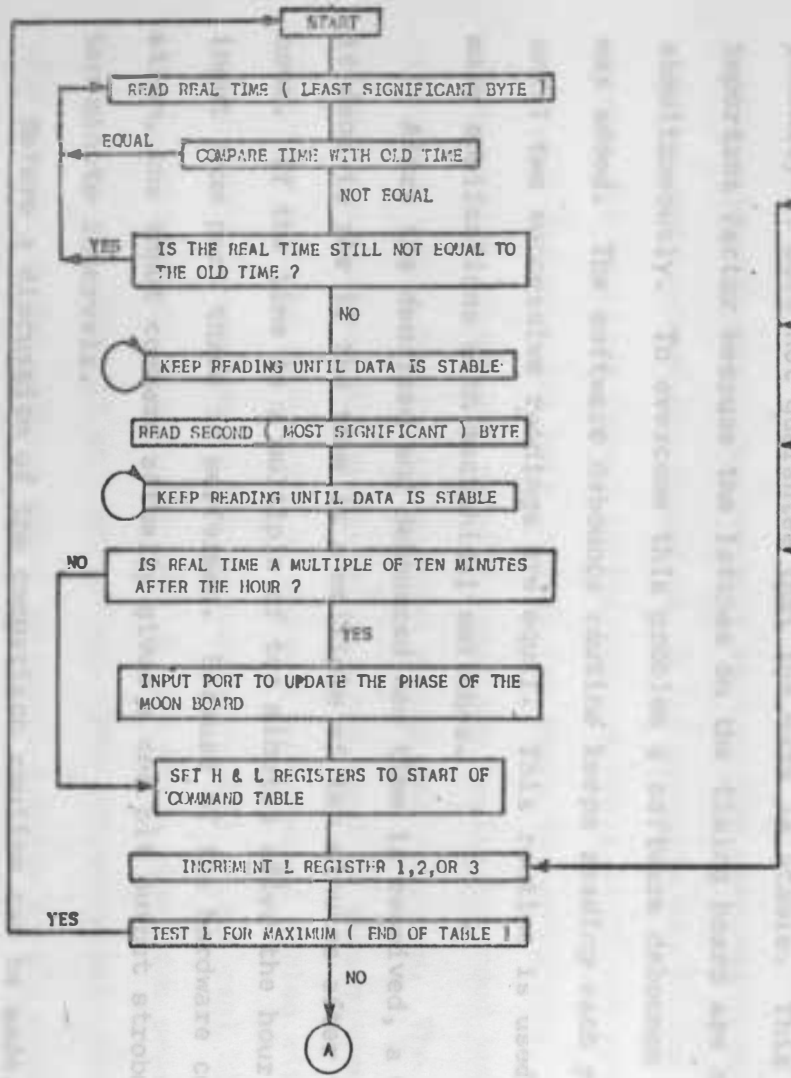
It was mentioned earlier that at this level of hardware and software it was impossible to totally separate the hardware and software discussions. In the discussion that follows on the program software there will be numerous references to the microcomputer

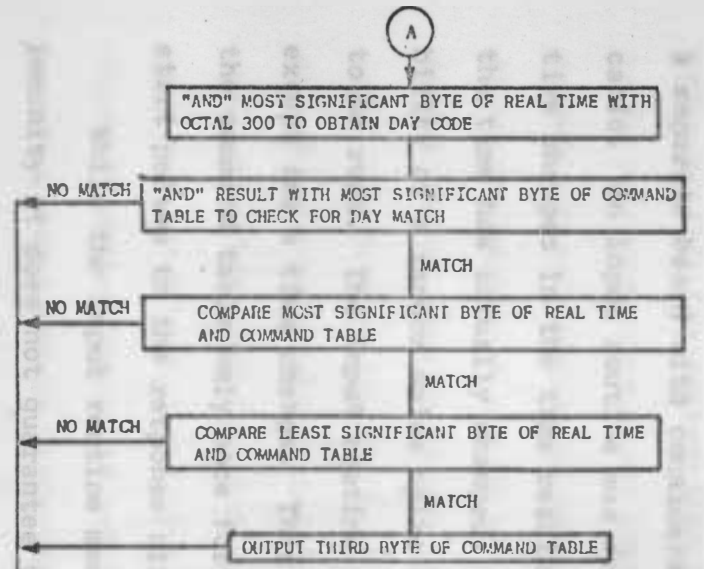
hardware. To understand the control program the requirements of the decision making section of the clock should be reviewed. The decision making section receives real time information from the timing section and compares it to a set of times in a command table. If the real time and command table times match, then the decision making circuitry outputs the appropriate command as shown in Figures 10 and 11.



FLOW CHART
FIGURE 10

The first task of the program is to obtain the day code and real time. While the real time and day code can be obtained with only two program statements accessing the two input ports, it was felt that a more sophisticated approach was needed. First, the two statement approach lacked noise immunity. Noise immunity was of





BIT	7	6	5	4	3	2	1	0
FIRST BYTE	WEEK- END	DAY	10 HOUR 2	1	HOUR 8	4	2	1
SECOND BYTE	NOT USED	10 MINUTE 4	2	1	MINUTE 8	4	2	1
THIRD BYTE	DAY	WEST	INTRUSION SET	RST.	POWER CH 1 S	R	CH 2 S	R

FLOW CHART & C.T. FORMAT

FIGURE 11

considerable concern since the real time information was derived on a separate board with considerable circuitry near the inter board cable. An input routine was developed that required two consecutive changes in the time reading from a previous time to show that the time has actually changed. This routine eliminated the possibility of a random noise spike on a line causing a spurious time to be read. The input routine also held the program in a read mode except for a time change. This has the effect of cycling through the command table only once for each new time eliminating multiple start pulses to the response circuitry.

While the input routine mentioned above improves the noise immunity it does not guarantee that the data is stable. This is an important factor because the latches on the timing board are not set simultaneously. To overcome this problem a software debounce routine was added. The software debounce routine keeps reading each port until two successive readings are equal. This routine is used in many applications with mechanical switches.

After the denoised and debounced new time is received, a check is done to see if the time is a multiple of ten minutes after the hour. If the time is a multiple of ten minutes after the hour an input from port three is performed. Because of the hardware configuration, the input command actually gives a one bit output strobe at ten minute intervals.

Before a discussion of the comparison routine can be made, the format of the command table must be understood. Each command is

listed in memory as a three byte command block. The first two bytes specify the appropriate time and type of day the command is to be executed. The third byte indicates which of the eight different commands are to be executed at the time indicated by the first two bytes. A bit by bit explanation of the command block format can be seen in Figure 11. The time in hours (twenty four mode) and minutes is in binary coded decimal format. Because significant bits in ten hours and ten minutes digits are uniformally zero, they have not been included in the command block. The day code section of the first byte differentiates the week into week days and weekend days and not into specific days of the week. This code was developed in conjunction with the hardware on the timing and overflow boards. The two bits sent to the computer from the timing section have the format of 10 for weekends and 01 for week days. The day code format in the command table is 10 for weekend only, 01 for week day only, and 11 for seven day operation. It can be seen that if the two sets of codes are logically anded and the result is zero, the output command should not be given. The most significant bit of the second byte is not used in this system, but could be used to differentiate between two command tables with a flag switch connected to the corresponding bit on the real time port. This feature could be used to advantage where two seasonal schedules are used in an institution. The output command byte, or third byte of the command block, determines which of the eight output lines are to be pulsed for a given day code and time match.

Setting any bit in the command byte to one will strobe the corresponding line during a match of the first two bytes. This information on the format of the command table should be of great assistance in understanding the remainder of the control program.

The comparison routine starts by setting the H and L registers to the memory address three bytes lower than the start of the command table. The H and L registers are used as a command table pointer throughout the comparison routine. The H register is uniformly zero because the end of the command table is less than address 256 in this application. The registers are set three bytes below the start of the command table and incremented three times so that the command table pointer is at the beginning of the command table. This indirect method of starting the command table reduces looping problems for testing each of the three byte command blocks as can be seen in Figure 11. After the command table pointer is incremented it is tested to see if it is beyond the end of the table. When the end of the command table has been reached, the program returns to the real time monitoring portion of the program. If the end of the command table has not been reached, the comparison routine is continued.

As has been mentioned earlier, the task of the decision making section of the clock is to compare the real time and day code with each of the command blocks in the command table. This comparison is actually accomplished in three stages. First a comparison is made of the real time and command table day code. This is accomplished

by masking off the hours from the day code and hours real time byte, leaving just the real time day code in the accumulator. This result is anded with the first byte of the command block and if this final result is not zero the day it appropriate for the command. The second stage of the comparison compares the remainder of the first byte of the command block, the ten hours and hours digits, to the real time ten hours and hours digits. The third stage of comparison, compares the real time ten minutes and minutes to the ten minutes and minutes in the second byte of the command block. If all three comparisons have matched, the third byte of the command block is sent to the output port and program returns to the L register increment point to examine the next command block.

If any of the three comparisons do not match, the program returns to the L increment point and the appropriate number of increments are performed to examine the next command block. When comparisons have been made on the entire command table the program returns to the real time monitoring portion of the program and waits for the real time to change. Readers that have worked with data processing programs regard infinite loop programs as a classical type of error in programming which causes failure to obtain the desired results. This is one example of the contrast between data processing programs where a terminal result is desired and real time control programs that yield a consistant pattern of behavior. From the flow chart in Figure 11 an assembly language program was written and hand assembled into octal machine code.[22] A listing

of the program in both assembly mnemonics and octal machine code along with the command table can be found in the appendix.

The octal program and command table code were placed in 1702A programmable read only memory using a control logic microcomputer and prom programmer. Several modifications were made in the program before the final performance requirements were met. The advantage of ultra violet erasable PROMS became very apparent when a chip could be erased and reprogrammed several times in an evening. This advantage is not without a cost. The field avalanche metal oxide semiconductor (FAMOS) technology requires a complex PROM programmer using a microcomputer to control its circuitry.[23]

After the program was properly placed into the 1702A, the chip was placed into the decision making system in the clock. Due to hardware problems in the input-output section of the computer board, the system did not work properly. It was decided to produce a short set of programs to find the hardware problems. The short programs found in the appendix were used to indicate problem areas. Further examination found faulty electrical contacts. The software diagnostics, although simple, were invaluable to fault location in the system.

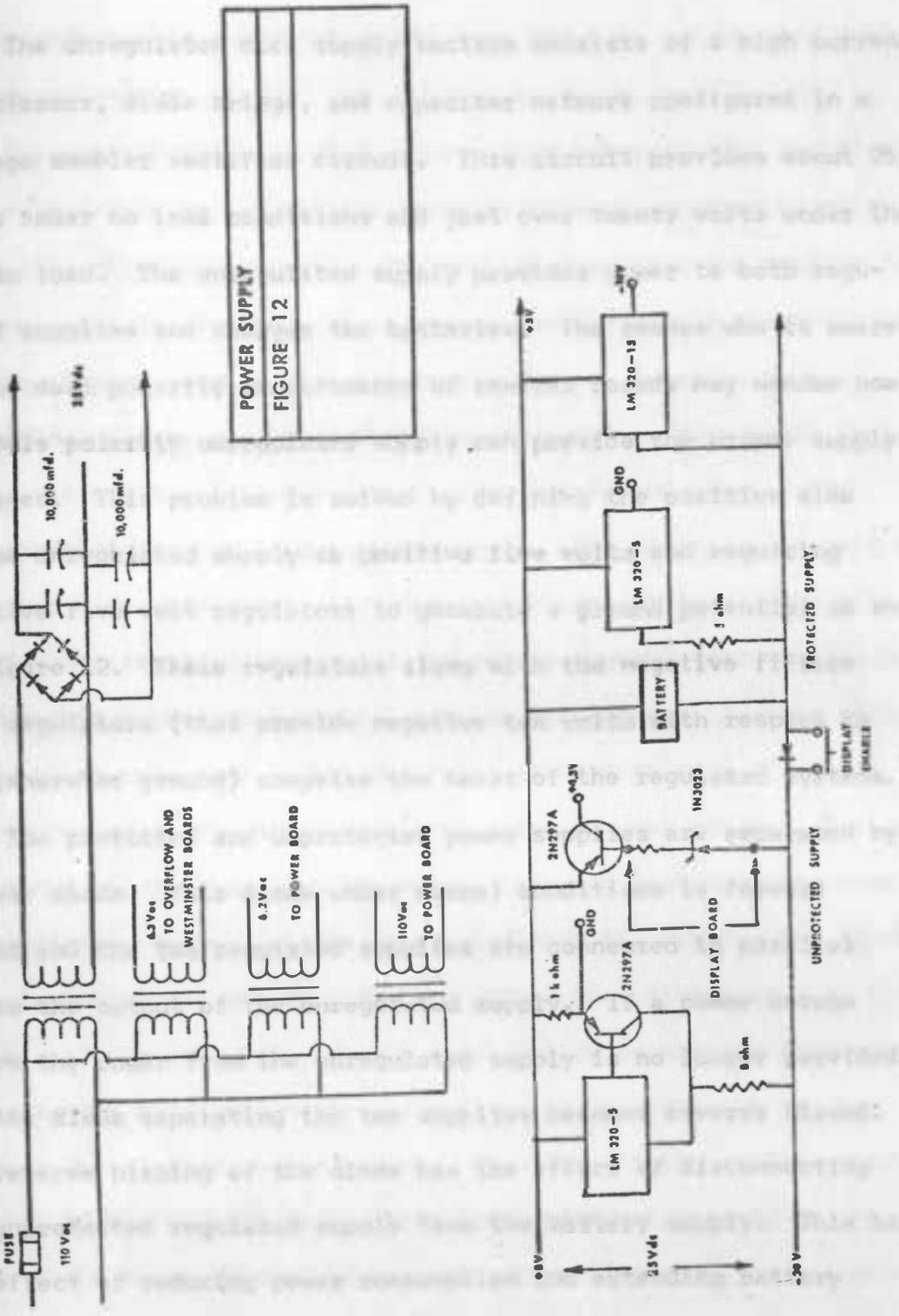
D. Power Source

Returning to the alarm clock analogy, another basic component is the power source. Early clocks simply had weights to drive the gearing systems. To make clocks portable, springs replaced weights. In the clock system under discussion the power supply design was

dictated by the requirements of the rest of the system and by the requirement of power outage immunity for certain sections of the clock.

The power supply can be divided into two major sections; the direct current power supply and the three transformers providing alternating current as can be seen in Figure 12. The 110 volt secondary transformer is used as an isolation transformer to reduce electrical shock hazards and is used only by the power board in the responding system. A 6.3 volt secondary transformer is also dedicated to the power board. This transformer provides power for an isolated on board power supply for the optical coupler network on the power board. This separate 6.3 volt transformer was used to completely isolate the high voltage circuitry on the power board from the rest of the system to protect the extensive circuitry on the other boards. Another 6.3 volt secondary transformer is used to power the Westminster melody board. The overflow board is also connected to the 6.3 volt output of this transformer to obtain the line frequency needed for the timing section. Details of the circuits mentioned can be found by consulting the discussion on the specific boards mentioned.

The direct current power supply provides power to almost all the digital logic in the system. The direct current power supply can be divided into three sections; the unregulated d.c. supply section, the unprotected regulated supply section, and a battery protected supply section.



POWER SUPPLY
FIGURE 12

The unregulated d.c. supply section consists of a high current transformer, diode bridge, and capacitor network configured in a voltage doubler rectifier circuit. This circuit provides about 25 volts under no load conditions and just over twenty volts under the system load. The unregulated supply provides power to both regulated supplies and charges the batteries. The reader who is aware of the dual polarity requirements of several boards may wonder how a single polarity unregulated supply can provide the proper supply voltages. This problem is solved by defining the positive side of the unregulated supply as positive five volts and requiring negative five volt regulators to generate a ground potential as shown in Figure 12. These regulators along with the negative fifteen volt regulators (that provide negative ten volts with respect to the generated ground) comprise the heart of the regulated systems.[24]

The protected and unprotected power supplies are separated by a power diode. This diode under normal conditions is forward biased and the two regulated supplies are connected in parallel across the output of the unregulated supply. If a power outage occurs the power from the unregulated supply is no longer provided and the diode separating the two supplies becomes reverse biased. The reverse biasing of the diode has the effect of disconnecting the unprotected regulated supply from the battery supply. This has the effect of reducing power consumption and extending battery life during a power outage. A push button switch was placed in parallel with the separation diode so that the unprotected circuitry

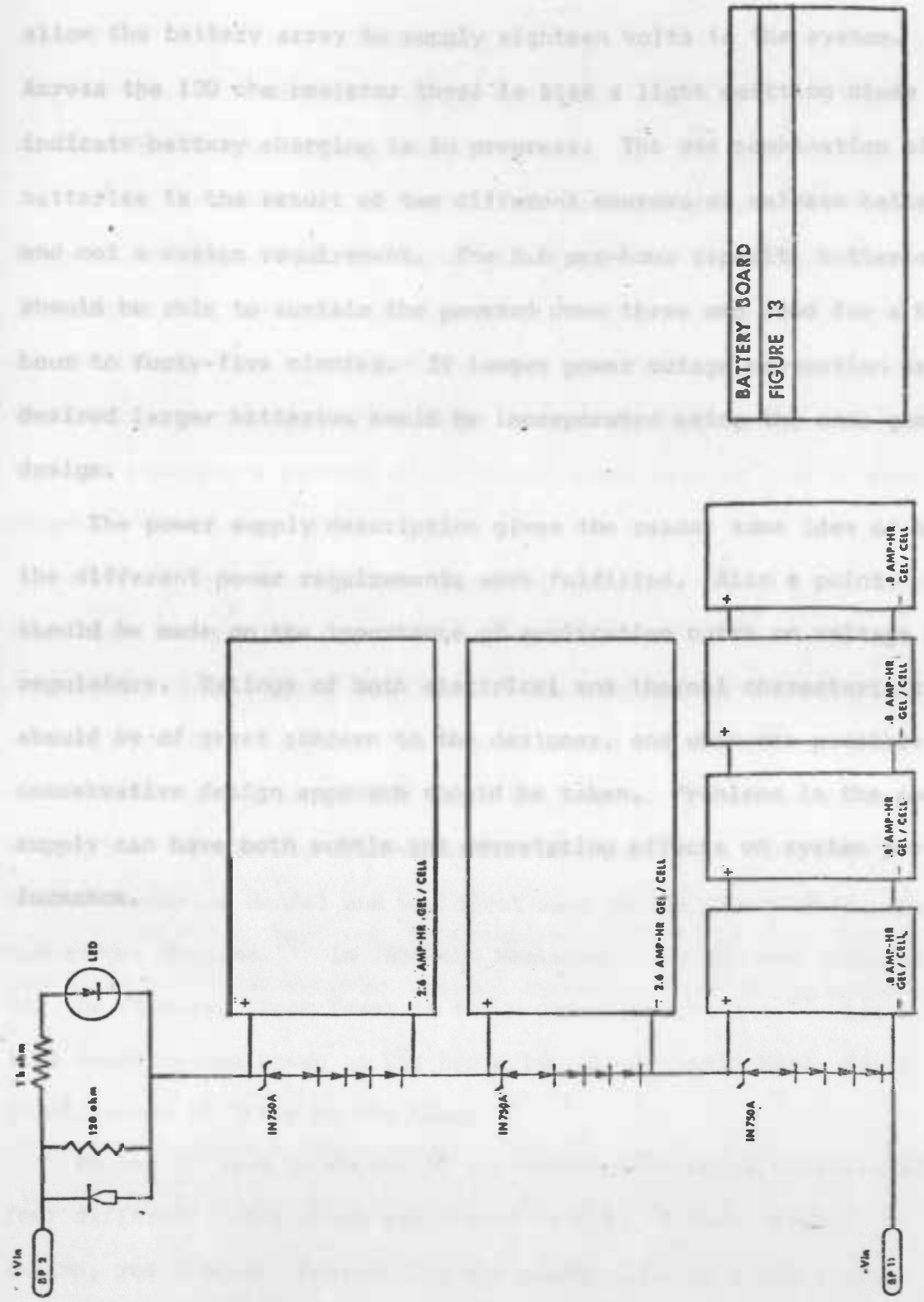
can be momentarily energized so the time and date can be read during a power outage.

The unprotected regulated power supply consists of a negative five volt regulator and emitter follower current boosting circuit and a line disabling circuit. The voltage regulator and emitter follower circuit provide a three ampere supply for the display board and the display section of the calendar board. During a power outage the voltage regulator floats up to positive five volts and forces the emitter follower circuit into cutoff and effectively disconnects this side of the power supply. The disabling circuit on the other side effectively disconnects the other side of the unprotected power supply during power outages to prevent current leakage through logic lines that could cause circuit damage. A zener diode and resistor were used in the disabling circuit to turn off the circuit faster than a simple resistor duing a power outage. For mechanical mounting reasons the zener diode and resistor were mounted on the display board. It was also found that the display had fewer thermal problems if ran on 4.3 volts rather than five volts, as the reduced voltage was used. When a power outage occurs the displays fade out over a ten second period and can be brought back by depressing the display enable switch. When the display enable switch is released the display fades again and continues to save the system about three amperes of current consumption during power outages.

The battery protected power supply consists of the voltage

regulator section and the battery board as shown in Figure 12. The voltage regulator section consists of five negative five volt regulators and three negative fifteen volt regulators connected as shown in Figure 12. It should be remembered that the negative five and fifteen volt regulators due to the circuit configuration yield the plus five and minus ten volt supplies that are required by the boards. Components that require negative nine volts rather than negative ten volts use diodes in the power supply line to drop the voltage to near the desired voltage. The five ohm power resistors that are in the supply lines of the negative volt regulators are used to reduce the voltage drop over the regulator and thus reduce thermal problems in the regulators. The regulators are, of course, attached to heat sinks and have capacitors attached from the reference line to the input and output lines for stability and denoising purposes.

The battery board is composed of six rechargeable Gel Cell six volt batteries and a set of components to limit both current and voltage during the charging of the batteries as shown in Figure 13. During charging the voltage across the batteries is limited to 6.8 volts by the zener and standard diode network. This is within the 6.9 volt trickle charge voltage that can be applied to the batteries on a constant basis.[25] To protect the batteries and zener diodes from excessive charging current a 120 ohm resistor is placed in series with the batteries. Across this resistor is a high current diode that will short out the resistor during discharge periods and


BATTERY BOARD
FIGURE 13
LED
1 k ohm
120 ohm
+ Vin
BP 2
2.6 AMP-HR GEL / CELL
2.6 AMP-HR GEL / CELL
.8 AMP-HR GEL / CELL
.8 AMP-HR GEL / CELL
.8 AMP-HR GEL / CELL
.8 AMP-HR GEL / CELL
IN750A
IN750A
IN750A
-Vin
BP 11

allow the battery array to supply eighteen volts to the system. Across the 120 ohm resistor there is also a light emitting diode to indicate battery charging is in progress. The odd combination of batteries is the result of two different sources of salvage batteries and not a design requirement. The 2.6 amp-hour capacity batteries should be able to sustain the powered down three amp load for a half hour to forty-five minutes. If longer power outage protection is desired larger batteries could be incorporated using the same general design.

The power supply description gives the reader some idea of how the different power requirements were fulfilled. Also a point should be made on the importance of application notes on voltage regulators. Ratings of both electrical and thermal characteristics should be of great concern to the designer, and whenever possible a conservative design approach should be taken. Problems in the power supply can have both subtle and devastating effects on system performance.

CHAPTER III

Output response section

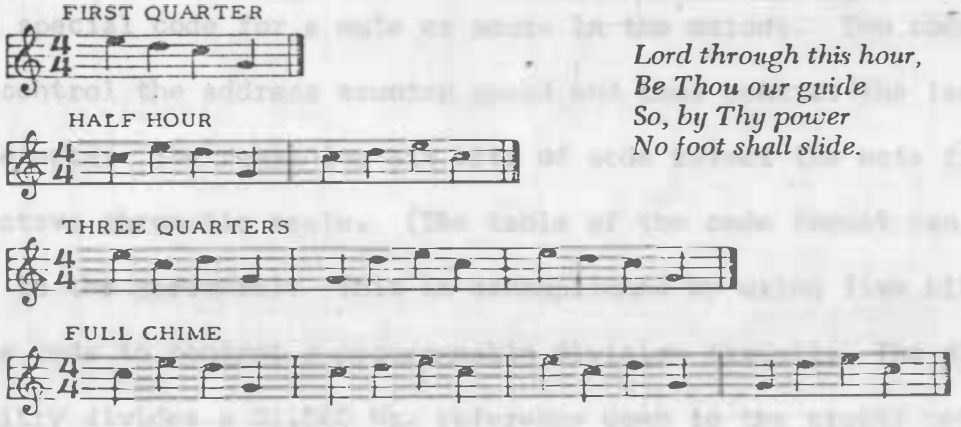
A. The Westminster Melody Section

While the previous discussion has centered around the time keeper, decision maker, and power supply, it is the output responding system that demonstrates the clock system's versatility and capabilities. The output modules that were developed demonstrate a variety of performance functions and design techniques. With the modular concept, a variety of different clock systems can be produced.

One of the most popular responding systems that has ever been developed for a clock is a bell system that rings a melody at uniform intervals. Among clock bell systems the Westminster chime, more commonly called "Big Ben," has become almost mandatory with long case (Grandfather) clocks and clock bell towers. The Westminster melody was arranged by William Crotch in 1793 from a symphony by George Frederick Handel and was first used on the tower of St. Mary's, Cambridge, England.[26] In 1859 the Westminster melody was selected for the Victoria Clock Tower in the House of Parliament.[27] Later this tower became known as Big Ben after Sir Benjamin Hall, First Commissioner of Works on the tower.[28]

As can be seen in Figure 14 the Westminster chime consists of four different notes which are played in sets of four, eight, twelve, and sixteen representing the quarter, half, three-quarter and full hour respectively. It is also important to note from

Figure 14 that through an hour cycle a pattern of twenty notes is repeated twice. With this information a module compatible with the digital clock system was developed that would play the Westminster melody on the quarter hour. The task of developing the Westminster module was divided into two sections, a melody section and a control section.

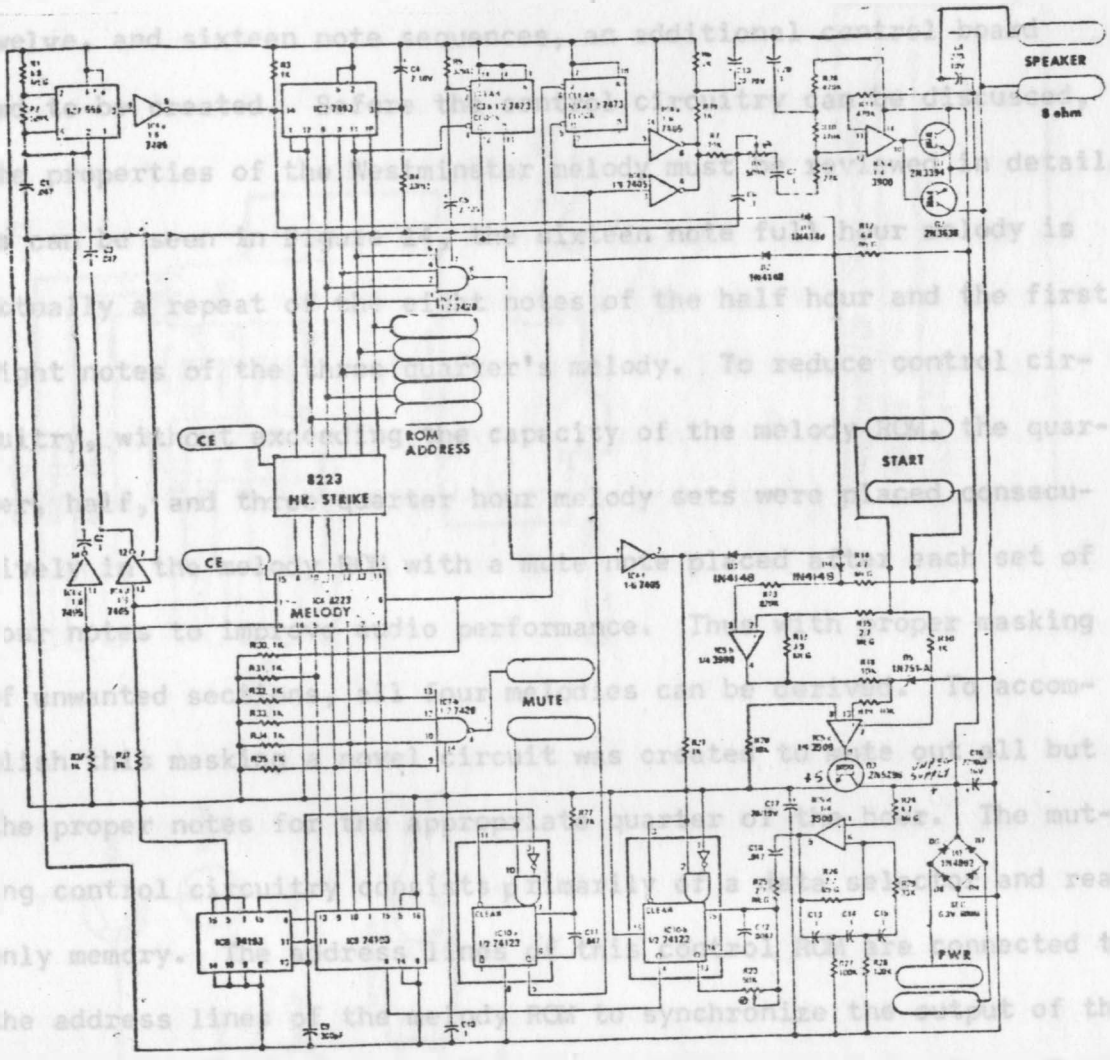


WESTMINSTER MELODY
FIGURE 14

The melody section has the responsibility of producing the twenty-four note Westminster melody, and the control section has the responsibility of selecting the appropriate portion of the Westminster melody corresponding to the quarter hour. While a totally original design was being developed for the melody section, a design for a solid state music box was published by Ralph E. Cousins in the September 1975 issue of Popular Electronics. While this design needed modification, it was decided to incorporate this

this system into the Westminster module to reduce design and construction time. The heart of the music box circuit is a Signetics 8223 read only memory that contains a code for the melody as shown in Figure 15. When the circuit is activated, a counter steps the memory through its thirty two addresses where each address represents either a musical note or a pause. The eight bit output code from the memory provides information for both pitch and note length and a special code for a mute or pause in the melody. Two code bits control the address counter speed and thus control the length of the note. The remaining six bits of code select the note from a two octave chromatic scale. (The table of the code format can be found in the appendix). This is accomplished by using five bits of the code to control a programmable division circuit. The division circuitry divides a 21,560 Hz. reference down to the proper note within the octave. The output of this division circuit along with a muting flag bit are supplied to a pulse widening circuit. If the muting flag bit is low, the output of the pulse widening circuit is kept low and no sound is transferred to the audio output circuitry, as can be seen in Figure 15. The remaining single code bit selects the appropriate octave by controlling a divide by two circuit. The audio output circuitry consists of an operational amplifier, transistor, push pull circuit, and an eight ohm speaker.

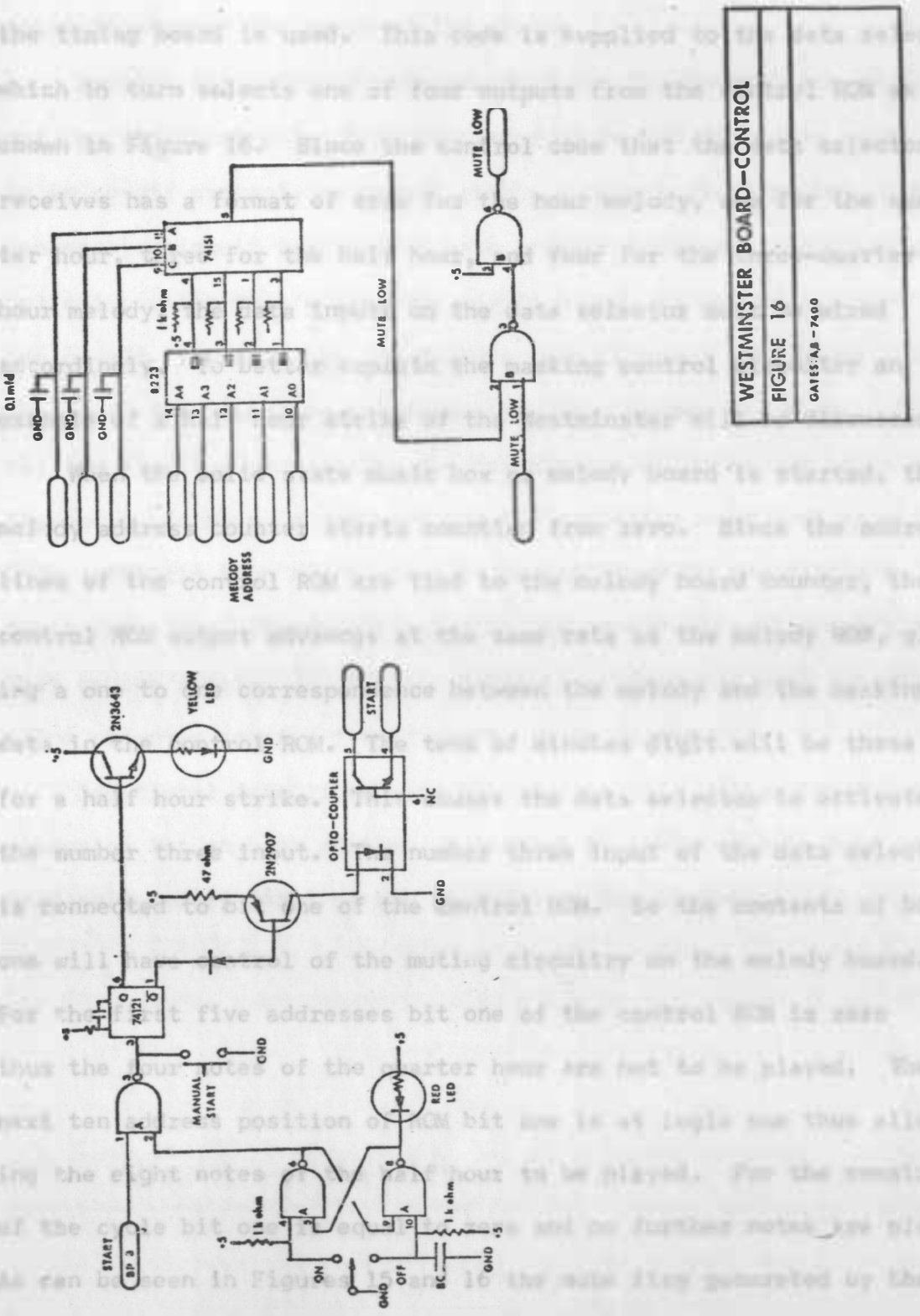
The solid state music box, as originally designed, would cycle through the entire thirty two note melody when the start switch was closed. To make this system perform the required four, eight,



WESTMINSTER BOARD—MELODY

FIGURE 15

twelve, and sixteen note sequences, an additional control board had to be created. Before the control circuitry can be discussed, the properties of the Westminster melody must be reviewed in detail. As can be seen in Figure 14, the sixteen note full hour melody is actually a repeat of the eight notes of the half hour and the first eight notes of the three quarter's melody. To reduce control circuitry, without exceeding the capacity of the melody ROM, the quarter, half, and three quarter hour melody sets were placed consecutively in the melody ROM with a mute note placed after each set of four notes to improve audio performance. Thus with proper masking of unwanted sections, all four melodies can be derived. To accomplish this masking a novel circuit was created to mute out all but the proper notes for the appropriate quarter of the hour. The muting control circuitry consists primarily of a data selector and read only memory. The address lines of this control ROM are connected to the address lines of the melody ROM to synchronize the output of the control ROM with the melody ROM to attain the proper masking. Four bits of the control output are used to determine the four masks that are needed to provide the four different melodies throughout each hour. The output of the control ROM is set to one if the note is to be played, and zero for unwanted notes in each of the four control masks. A data selector was used to select the proper mask corresponding to the desired melody. The desired melody is a function of the time within the hour. To obtain the necessary information for melody selection, the three bit code of the ten minutes digit from


WESTMINSTER BOARD—CONTROL
FIGURE 16
GATES: AJ - 7400
MUTE LOW
MUTE LOW
MUTE LOW
0.1 mfd
GND
GND
GND
8223
A4
A3
A2
A1
A0
MELODY ADDRESS
74151
1 k ohm
+5
2N3643
YELLOW LED
GND
START
OPTO-COUPLER
NC
47 ohm
2N2907
GND
74121
MANUAL START
GND
+5
RED LED
1 k ohm
1 k ohm
START
BP 2
ON
OFF
GND
BC
GND

the timing board is used. This code is supplied to the data selector which in turn selects one of four outputs from the control ROM as shown in Figure 16. Since the control code that the data selector receives has a format of zero for the hour melody, one for the quarter hour, three for the half hour, and four for the three-quarter hour melody, the data inputs on the data selector must be wired accordingly. To better explain the masking control circuitry an example of a half hour strike of the Westminster will be discussed.

When the solid state music box or melody board is started, the melody address counter starts counting from zero. Since the address lines of the control ROM are tied to the melody board counter, the control ROM output advances at the same rate as the melody ROM, giving a one to one correspondence between the melody and the masking data in the control ROM. The tens of minutes digit will be three for a half hour strike. This causes the data selector to activate the number three input. The number three input of the data selector is connected to bit one of the control ROM. So the contents of bit one will have control of the muting circuitry on the melody board. For the first five addresses bit one of the control ROM is zero thus the four notes of the quarter hour are not to be played. The next ten address position of ROM bit one is at logic one thus allowing the eight notes of the half hour to be played. For the remainder of the cycle bit one is equal to zero and no further notes are played. As can be seen in Figures 15 and 16 the mute flag generated by the melody ROM is still functional and is used after each set of four

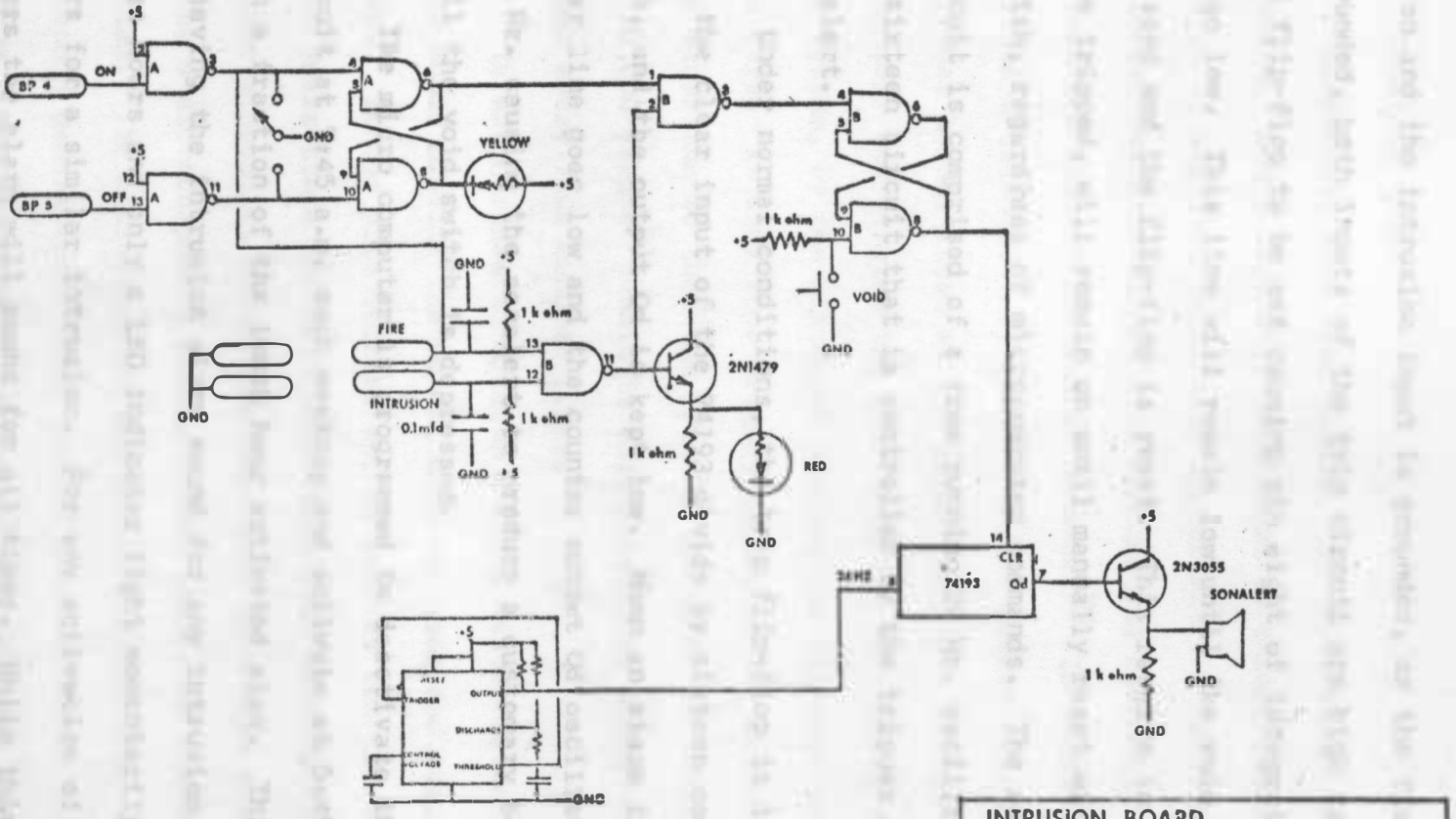
notes to improve the tempo of the system.

Another circuit that was developed for the Westminster system was the start circuit. This consisted of an S-R flip-flop used as an on-off control, an on-off light, a manual start switch, and a monostable circuit that drives both a start indicator light and an optical coupler that starts the melody board, as shown in Figure 16. The on-off flip-flop has the feature that if power is removed from the board the flip-flop automatically flips to the off state. This has the effect of cancelling the Westminster chime at odd hours if the power has been interrupted and the batteries have been exhausted. This automatic turn off feature is accomplished by connecting a large capacitor between off input of the on-off flip-flop and ground. This causes a delay in off line going to a logic one with the restoration of power causing the flip-flop to be turned off.

Additional data lines were provided in this design for future expansion for an hour strike. A listing of cable designations and read only memory listings can be found in the appendix.

B. Intrusion Board

Intrusion and fire detection equipment is becoming an increasingly important application area for digital electronics. To demonstrate how a small amount of circuitry can fulfill these security requirements, an intrusion board module was created for the clock system. No attempt was made to design new detection transducers, but rather to design circuitry that could use existing transducers. As can be seen in Figure 17, the intrusion board consists of an on-off flip-flop and indicator, a transducer input and indicator, an alarm trip circuit, and an audio output circuit. The on-off flip-flop is simply an S-R flip-flop that can be controlled by a positive pulse from the microcomputer, manually set or reset, or set by the fire transducer circuitry. The on-off flip-flop lights a yellow indicator and activates the trip circuit by driving pin one of integrated circuit B high. The transducer input circuit consists of a NAND gate with pullup resistors on its input which causes this output to be low except when one transducer grounds either the fire or intrusion inputs. The intrusion input only affects this gate while the fire input, when grounded, also turns on the on-off flip-flop enabling the system. This configuration has the effect of having the system respond to a fire transducer at all times while the intrusion system is only activated by either computer or manual activation for after hour protection. A red LED indicator circuit was connected to the input transducer gate to show the response of the intrusion board when the system is in an off state. When the intrusion system



INTRUSION BOARD
FIGURE 17
GATES: A, B – 7400

is on and the intrusion input is grounded, or the fire input is grounded, both inputs of the trip circuit are high causing the trip S-R flip-flop to be set causing pin eight of integrated circuit B to go low. This line will remain low until the void switch is depressed and the flip-flop is reset. This results in a system that, once tripped, will remain on until manually reset with the void switch, regardless of microcomputer commands. The audio output circuit is comprised of a free running 24 Hz. oscillator, a divide by sixteen circuit that is controlled by the trigger circuit and a sonalert.

Under normal conditions, the trip flip-flop is in a reset state and the clear input of the 74193 divide by sixteen counter is driven high, and the output Qd is kept low. When an alarm is tripped, the clear line goes low and the counter output Qd oscillates at about 1.5 Hz. causing the sonalert to produce a customary beep-beep sound until the void switch is depressed.

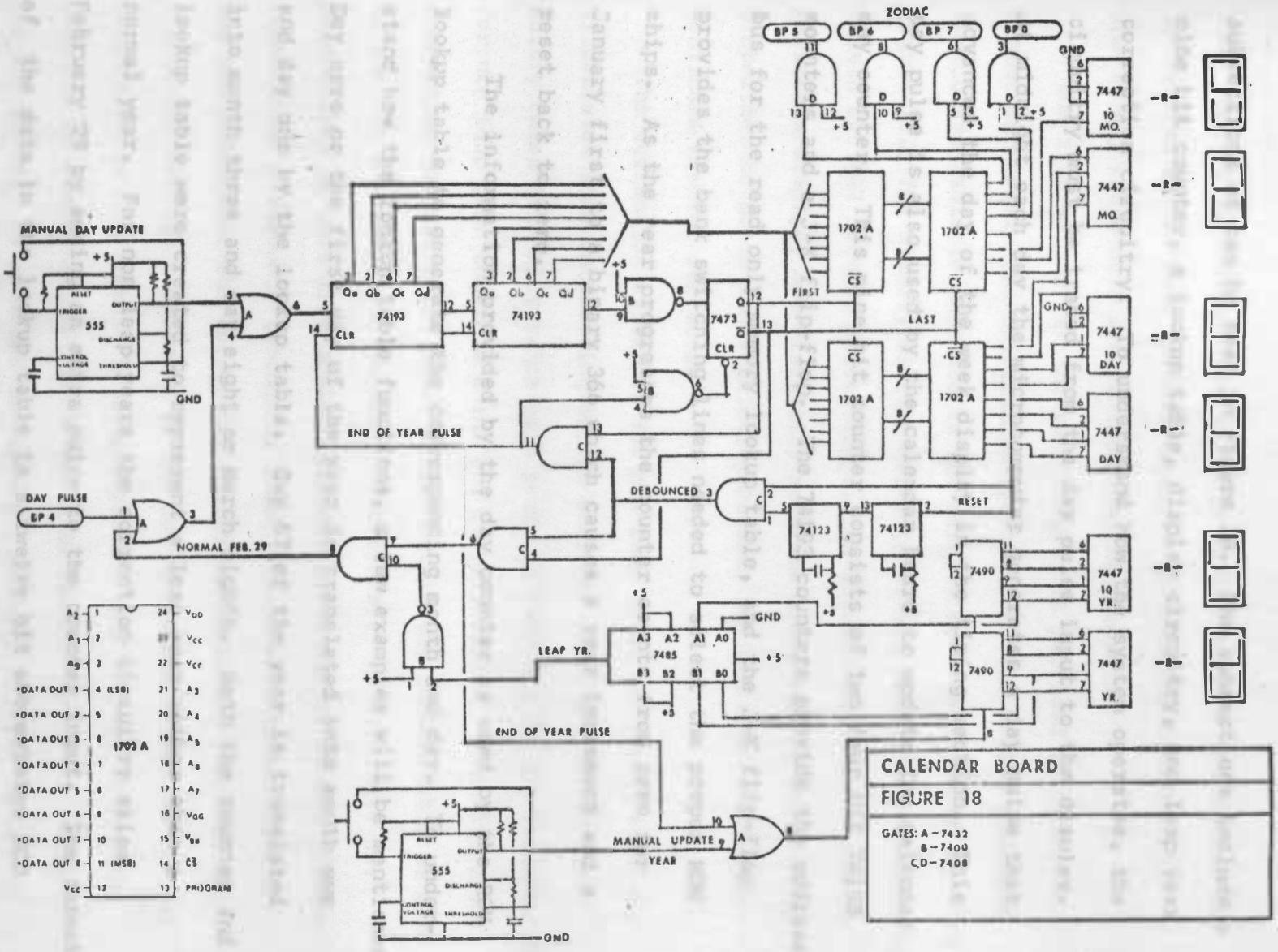
The micro computer is programmed to deactivate the intrusion circuit at 7:45 a.m. each weekday and activate at 5:45 p.m. each day, with a fraction of the lunch hour activated also. This has the effect of having the intrusion alarm sound for any intrusion during non business hours and only a LED indicator light momentarily during business hours for a similar intrusion. For any activation of the fire transducers the alarm will sound for all times. While this circuit is not overly complicated, it does illustrate a useful modular addition to a clock system for industrial use.

C. Calendar Board

Calendars, like time pieces, have interested men since the beginning. Early calendars were based on changing seasons or the behavior of animals, which only approximated a uniform cycle. Lunar calendars were developed from the 364 day cycle but this led to errors and corrections had to be made.[29] In 46 B.C. Julius Caesar created a calendar very similar to the modern calendar.[30] This calendar became known as the Julian calendar and provided for a leap year and started the year on March 1. Due to slight errors in the system, by the sixteenth century the calendar was in error by ten days. In 1582 Pope Gregory corrected the calendar and improved the algorithm so that the present system is in error only one day in 20,000 years.[31] This is accomplished by skipping leap year three times every four hundred years.

It was decided to develop a solid state calendar board module that would provide the month, day, and year and correct for leap years. It was also decided not to correct for the Gregorian correction years since the next correction year is 2100 A.D. Three different approaches could have been used to create a calendar board: using a microcomputer, purely random logic, or a counter table lookup system. The microcomputer approach would show the most promise in a second generation system, but it was decided to use the counter lookup table arrangement because of its relative ease of implementation and conservative design.

The calendar board that was designed consists of several



CALENDAR BOARD

FIGURE 18

GATES: A – 7432
B – 7400
C,D – 7408

subsections as can be seen in Figure 18. The subsections include a nine bit counter, a lookup table, display circuitry, and leap year correction circuitry. To understand how the system operates, the circuitry must be traced from the day pulse input to the display. At midnight each day the microcomputer generates a day pulse that advances the day of the week display in the timing section. This day pulse is also used by the calendar board to update the calendar day counter. This nine bit counter consists of two four bit 74193 counters and a J-K flip-flop. The 74193 counters provide the address bus for the read only memory lookup table, and the J-K flip-flop provides the bank switching lines needed to select the proper ROM chips. As the year progresses the counter counts from zero for January first to a binary 366 which causes a year increment and a reset back to zero.

The information provided by the day counter is used by the ROM lookup table to generate the corresponding month and day. To understand how the lookup table functions, a few examples will be mentioned. Day zero or the first day of the year is translated into month one and day one by the lookup table. Day 67 of the year is translated into month three and day eight or March eighth. Both the counter and lookup table were created to represent a leap year rather than a normal year. For non leap years the correction circuitry skips February 29 by adding an extra pulse to the counter input. The format of the data in the lookup table is a twelve bit abbreviated BCD code for the month, day, and reset flag and four bits that can be

user defined to indicate special days of the year or the houses of the zodiac as was done in this example. A detailed listing of the lookup table data can be found in the appendix.

The information for the year display is provided by two 7490 decade counters. The information generated by the decade counters and lookup table is supplied to the display section as shown in Figure 18. The display section consists of six common anode circuits driven by 7447 decoder driver chips. An attempt was made to multiplex this display, but visibility problems arose due to solid state properties of green light emitting diodes, and the discrete display mentioned earlier was used.

The correction circuitry for non leap years mentioned earlier, illustrates some interesting design schemes. The correction circuitry comes into play twice each year, on February 29 and December 32 for non leap year correction and end of year reset. To activate the correction circuitry the reset output bit in the lookup table is set to one for the two dates previously mentioned. To eliminate the possibility of this line giving spurious reset pulses during address transitions a debounce circuit is used. To differentiate between the two tasks that the correction circuitry performs, the most significant bit and its complement of the day counter are used to control which task is performed. To understand the end of year reset, the sequence of events at the end of the year must be covered. When the day counter is advanced to day 366 (December 32) the output of the reset bit goes to a logic one and causes the debounce circuit

to issue a pulse. This pulse is routed to clear the day counter which has the effect of jumping to January first. The routing of the debounced reset pulse is controlled by the most significant bit of the day counter which has the effect of enabling the leap year circuitry for the first part of the year and the end of the year reset circuitry for the last part of the year. The end of the year pulse, used to clear the day counter, is also used to increment the year counter so that the year display is also updated at the end of the year.

The task of the leap year circuitry is to determine whether or not the year displayed is a leap year and set the length of February accordingly. The leap year recognition circuit uses a unique algorithm developed by the author. Unlike the standard leap year test of examining the year for divisibility by four, this method compares data bits within the BCD year code. To understand how this bit pattern algorithm was derived, one must observe the pattern of leap years in even and odd decades. In even decades the last digit of a leap year is either a zero, four, or eight. In odd decades the last digit of a leap year is either a two or a six. The information just mentioned gives rise to two rules for a leap year test. First, all leap years are even years, that is the least significant bit of the least significant digit must be zero for the year to be a leap year. The second rule is somewhat less obvious. Among even years of both even and odd decades the least significant bit of the ten year's digit is equal to the second least significant bit of the

year's digit, for leap years only. Using these two rules and a 7485 magnitude comparitor chip a leap year flag can be generated.

To understand how the remainder of the leap year circuitry works the sequence of events that occur when the day counter reaches February 29 must be observed. When the day counter reaches 59 the lookup table outputs the code for February 29. Included in this output is the reset output which is at a logic one for this date. The reset output causes the debounce circuit to produce a delayed pulse. This pulse is routed to the leap year circuitry due to the most significant bit of the day counter being equal to zero. This routing is accomplished by logically "ANDing" the debounced reset pulse with the complement of the most significant bit of the day counter. This result is further "ANDed" with the complement of the leap year flag as shown in Figure 18. This results in the addition of an extra day pulse on February 29 for non leap years causing the day counter to advance and have March first displayed. What is seen on the display at the end of February 28 during a non leap year is first a turn to February 29. After approximately half a second it automatically advances to March first.

As can be seen in Figure 18, two switch controlled oscillators were added to the system to manually adjust the date and year. These oscillators were set at about two hertz for ease of adjustment. Four "AND" gates were also added to buffer the outputs of the user designed bits of the lookup table to reduce the danger of static electricity damage to these lines that leave the board. As was

mentioned earlier, the user defined bits were programmed with houses of the zodiac, one for Aries through twelve for Pisces. The display for this information is on the phase of the moon board and will be discussed further in conjunction with that board. Another point about the calendar board is the separation of the display power supply from the rest of the board. Since the common anode current of the display is significant and not necessary for data retention, it was decided to power this section from the unprotected power supply to extend battery life during power outages. This display, like the main display, can be momentarily read during a power outage by pressing the display enable switch.

The previous discussion of the calendar board should give the reader an idea of how a counter and lookup table can be used to advantage to provide information that is not easily derivable through Boolian equations. Also, the reset and correction circuitry illustrates some useful algorithms based on bit patterns rather than traditional definitions. In summary, the calendar board provided both an interesting circuit design task and an interesting end result.

D. Phase of the Moon Board

One of the earliest additions to a mechanical clock dial was the lunar or moon dial that gives the current phase of the moon. There have been many reasons for desiring this information. First, many early calendars were based on the $29\frac{1}{2}$ lunar cycle and these calendars are used to determine important religious holidays such as Passover, Easter, and the religious calendar of Islam.[32] Sea captains also used the phase of the moon to calculate the approximate time of high and low tide. High tides generally occur when the moon is at the zenith.[33]

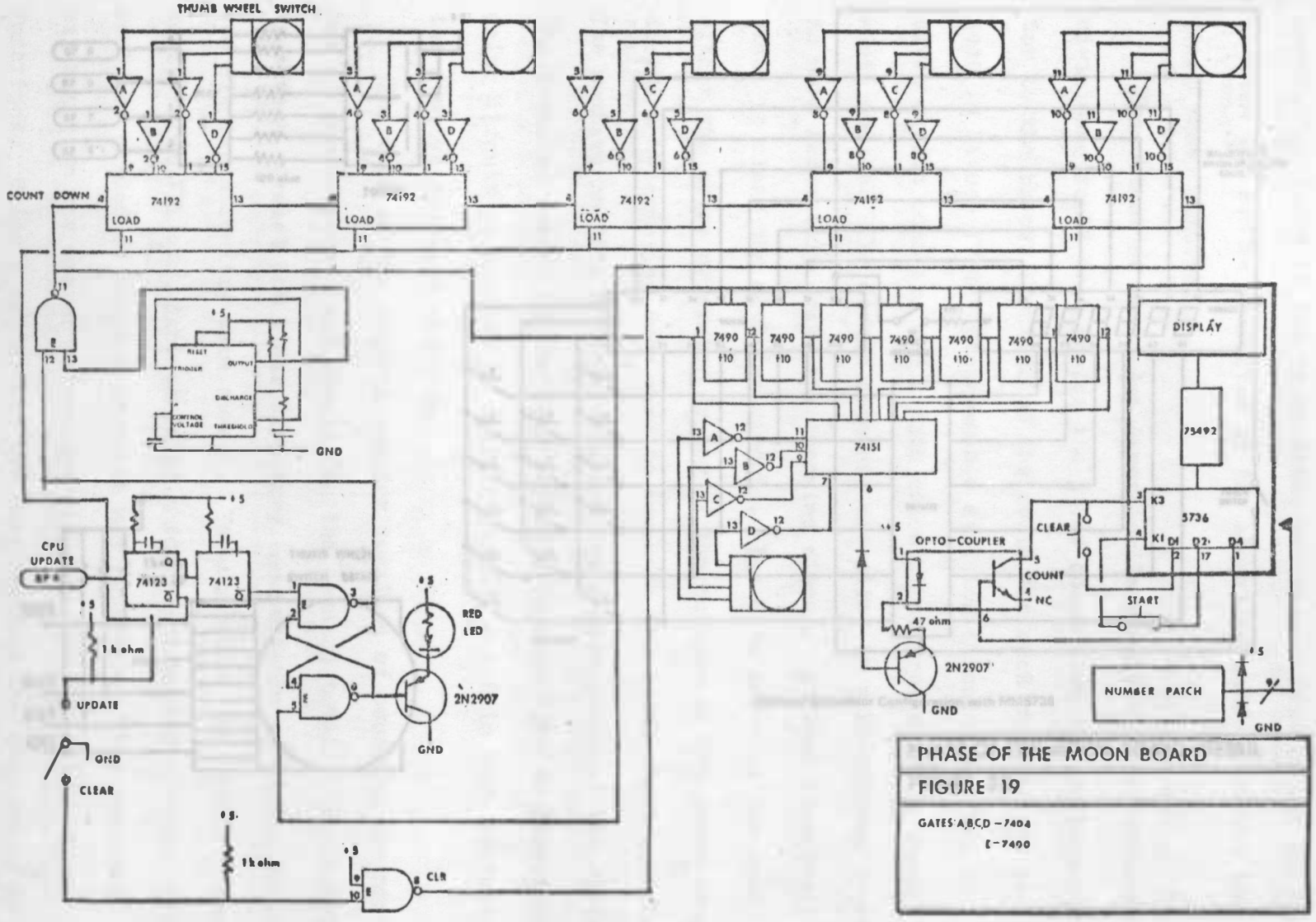
Moon dial displays in mechanical clocks usually take the form of a geared wheel placed behind a stationary lunar arch face plate with a semicircular opening. The front of the geared wheel has two stylized moon faces placed on opposite halves with either a scenery or star background. This wheel is rotated once every two lunar months or 59 days to show the phases of the moon using one of the moon faces at a time. To attain the 59 day period of rotation of the moon wheel, 118 gear teeth were cut in the wheel so as to allow the hour gears to advance the moon wheel one tooth every twelve hours. This gearing approach provided the fundamental concept for a digital logic lunar dial or what came to be known as the phase of the moon board.

When it was first decided to develop a lunar dial, several non-numeric displays were considered and rejected. After a discussion

with David Edward Graen, Watertown, South Dakota, an erudite of esoterica, it was decided to define the lunar cycle from zero to one on a digital display, or more specifically a new moon is 0.00, a first quarter is 0.25, a full moon is 0.50, last quarter is 0.75, and the next new moon is 1.00.

To emulate the mechanical gearing system on the moon wheel a digital system consisting of a pulse generator, division circuit, and counter was needed. While this system could have been implemented in hard wired random logic, it was decided to produce a programmable device using some novel circuit approaches.

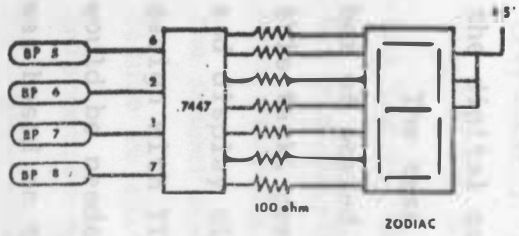
The pulse generator, as shown in Figure 19, has the task of providing a predetermined number of pulses each time it is strobed with an update pulse. This task was accomplished by using an oscillator, presettable counter, and control circuitry. When an update pulse is received by the pulse generator, a monostable multivibrator loads the presettable counters with the data from the thumb wheel switches. A detail of the thumb wheel switch can be seen in Figure 20. A flip-flop is then set by a second monostable which enables the output of the oscillator and turns on a LED indicator. The oscillator proceeds to clock the counter in a count down mode from the preset number. When the count reaches zero, a borrow pulse is sent from the most significant digit counter to the flip-flop causing the oscillator to be disabled. A detailed study of race conditions and timing diagrams indicates that the number of pulses generated by this pulse generator is actually one more than the

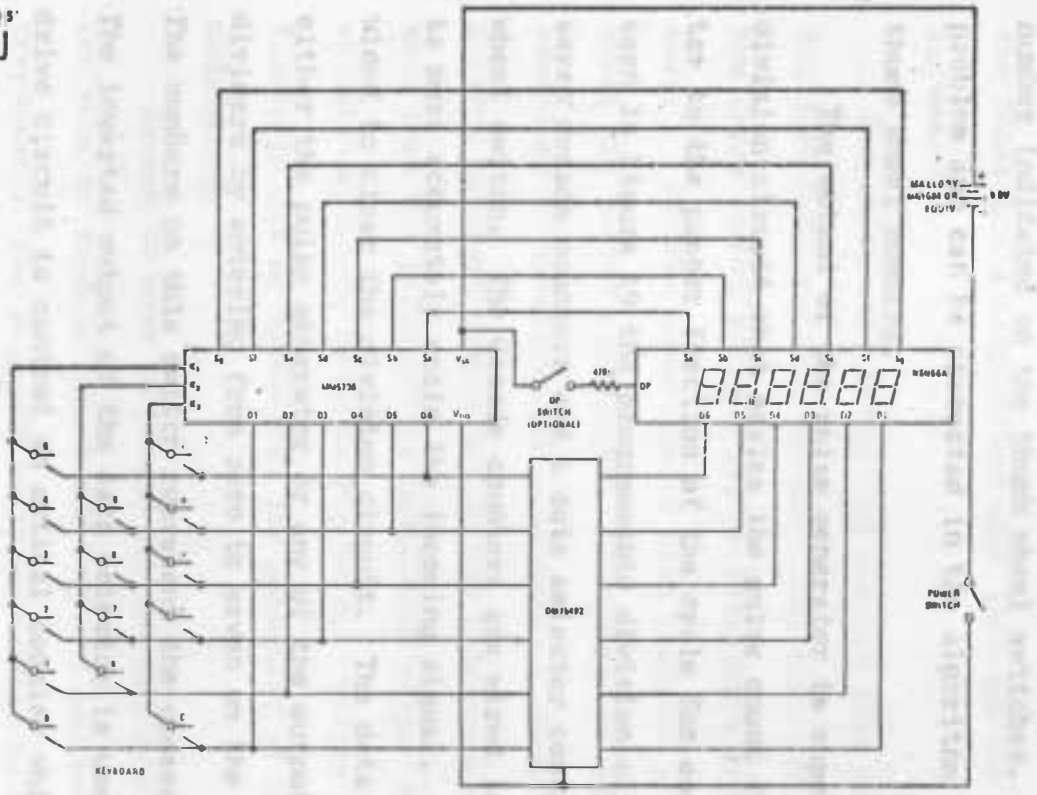


PHASE OF THE MOON BOARD

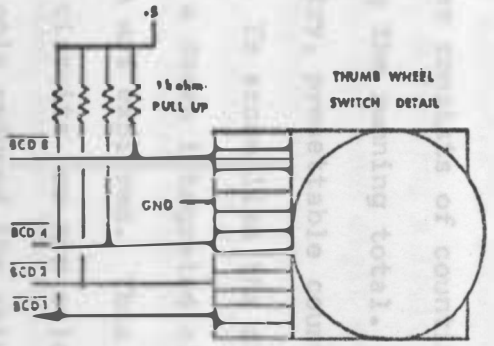
FIGURE 19

GATES A,B,C,D – 7404
E – 7490





Optional Calculator Configuration with MM5736



PHASE OF THE MOON BOARD--DETAIL

FIGURE 20

number indicated on the thumb wheel switches. This causes no real problem and can be corrected in the algorithm used to derive the thumb wheel numbers.

The output of the pulse generator is supplied to a programmable division circuit that scales the pulse count from the pulse generator to the proper fraction of the cycle for each update. As can be seen in Figure 19 the programmable division circuit consists of seven decade counters and a data selector controlled by a thumb wheel switch. The decade counters are wired in a bi-quinary format to more accurately scale the incoming signal. A clear line was provided to clear the division circuit. The data selector can read either the pulse generator or any of the outputs of the seven decade dividers by entering from zero to seven on the thumb wheel switch. The numbers on this switch represent the chosen power of ten division. The inverted output of the data selector is used with an inverting drive circuit to control an optical coupler which in turn controls the digital counting circuit.

The task of the digital counter consists of counting the number of scaled pulses and displaying the running total. To accomplish this task, preset and clear circuitry, presettable counters, drivers, and display circuitry were needed. To accomplish the digital counter design with TTL circuitry at least a dozen integrated circuit chips would be needed so another approach was explored. This approach was based on the use of calculator circuits for non calculator functions. Using a calculator not only reduced the chip count to

two chips and a display but also improved the versatility of the system.

The calculator integrated circuit that was selected was the National Semiconductor MM5736. This chip was selected because of the large amount of application literature and documentation available.[34] Another advantage of this chip was its presence in the NOVUS 650 pocket calculator. Obtaining an entire calculator had the advantage of simultaneously obtaining a compatible display, display driver, keyboard number patch, and interconnect board for little more than the price of the calculator chip. A schematic of the calculator can be found in Figure 20. Before the calculator modifications can be discussed, the operation of the calculator must be understood.

As can be seen in Figure 20, the MM5736 calculator chip has two sets of outputs (digit and segment) and one set of inputs (keyboard). The chip pin count is reduced by multiplexing both the display and keyboard. The display multiplex is accomplished by enabling the six digits one at a time by outputting a logic one on one of the six display enable lines, D1-D6. For each digit enabled, the segment drivers, Sa-Sg, activate the appropriate segments to represent the proper numerical value for that digit. The keyboard as shown in Figure 20 also makes use of the digit enable lines to input keyboard information through lines K1-K3.

When a keyboard contact is closed, one of the keyboard input lines and one of the digit enable lines are connected. The internal

decode circuitry decodes and denoises this information and responds accordingly. The denoising circuitry puts an upper limit on the speed of the calculator response, but does not cause major difficulty in this application. Understanding the format of the keyboard input both from an electrical and key entry point of view is necessary to properly design the interface circuitry between the calculator and the rest of the system. The electrical requirements for the keyboard contacts are relatively simple in that there are no debounce requirements, but open contacts are required to be much greater than 50 kilo-ohms. This high resistance requirement requires the optical couplers to be used in a photo diode rather than the more common photo-transistor mode for keyboard interface as shown in Figure 19.

The key entry format for the MM5736 calculator conforms to the reverse Polish method of entering the two operands and then the operator, but it differs from the advanced calculators in that there is no register role down. This has the effect of adding the second operand to the first operand or previous result each time the ENT + key is pressed, so if the second operand is one and the + key is connected to the external system a digital counter is created.

The specific interface that was used in the phase of the moon board consisted of two normally open toggle switches to initialize the counter and an optical coupler to increment the counter. The switches are connected to perform the functions of clear and the entry of one or start, as shown in Figure 19 and Figure 20.

The optical coupler is connected to perform multiple closures of the + switch or successive additions of one to the running total. It was also decided to provide an on board connector so that a complete keyboard can be temporarily utilized to initialize the system at some point other than zero. This keyboard feature also enables the system's other features such as counting down and incrementing by a number other than one. To protect the calculator chip from static electricity damage, a set of eighteen diodes were connected to the keyboard connector lines to limit voltage swings on these lines as shown in Figure 19.

In any programmable device such as the phase of the moon board, the hardware cannot function without being programmed. The programming of this board consists of two parts, the setting of the cycle length on the thumb wheel switches and the initialization of the cycle with the switches and keyboard.

Programming the phase of the moon board for the proper cycle duration involves several algebraic operations and entry of the results into the thumb wheel switches. Since the system is updated every ten minutes, a calculation must be performed to determine the fraction of a cycle that constitutes ten minutes. This can be accomplished by simply dividing the ten minutes period by the cycle length in minutes, and leave the quotient in scientific notation. From this information the pulse generator can be programmed. This is done by entering the five most significant digits of the quotient on the thumb wheel switches. At this point it should be remembered

that the pulse generator produces one more pulse than the total indicated on the switches so the total number on the switches should be decremented by one to correct for this factor. To program the division circuitry the magnitude scaling number can be derived indirectly from the exponent of the quotient derived earlier. While the derivation of the following rule is involved the rule itself is simple: take the exponent, subtract two and enter the number on the thumb wheel switch for the division circuit.

To better explain the method two examples will be discussed. First, the moon phase cycle, which has a cycle length of 29 days, 12 hours, 44 minutes, and 2.8 seconds will be covered. This length of time translates into 42,524.0466... minutes. Dividing ten minutes by the cycle length just calculated yields the quotient 2.351610 x 10 - 4. The first five numbers can now be entered into the thumb wheel switches remembering that the total number must be decremented by one so the thumb wheel switches would indicate 23515. To program the division circuitry from the quotient, two must be subtracted from the exponent, or -4 -2 = -6, and then the scaling number, six, can be entered into the remaining thumb wheel switch to complete the cycle length programming for the phase of the moon.

Programming the board to monitor the cycle of the tides can be done in a similar manner. First the 12 hour and 24 minute tide cycle can be translated into 744 minutes. Dividing ten minutes by 744 minutes yields a quotient of 1.3441×10^{-2}. Decrementing the set of five significant number by one, gives the result of 13440 that is

entered on the thumb wheel switches. To program the division circuitry, two units are subtracted from the quotient exponent and the result, -2 -2 = -4, or four is entered on the last thumb wheel switch. Using this method of cycle programming the phase of the moon board could be used to monitor any cycle from an hour to several years limited only by the division circuitry which has a scale number range of from two to seven.

After the cycle length has been programmed the calculator digital counter must be initialized. This is accomplished by pressing the toggle switch to clear the division circuitry and pressing the toggle switch to clear the calculator. At this point, if a starting point other than zero is desired the number patch of keyboard can be plugged into the board and the desired starting point can be entered and the keyboard removed. Then the start toggle switch can be depressed to enter a one for the increment number and then the system is ready to monitor the appropriate cycle.

Another constituent of the phase of the moon board that is independent of the rest of the board is the Zodiac or special day display. This display receives its information from the calendar board as shown in Figures 18 and 20, and uses a standard 7447 common anode display circuit. The display has the capability of displaying the digits 0-9 and four non numeric symbols for 10-14. For the zodiac the digits 1-9 and the symbols for ten, eleven and twelve are used to represent the twelve houses of the Zodiac. A code table can be found in the appendix.

The versatility of this board is such that its applications are limited only by the imagination of the user. The phase of the moon board has been the most interesting from both circuit and use stand point of all the output response boards.

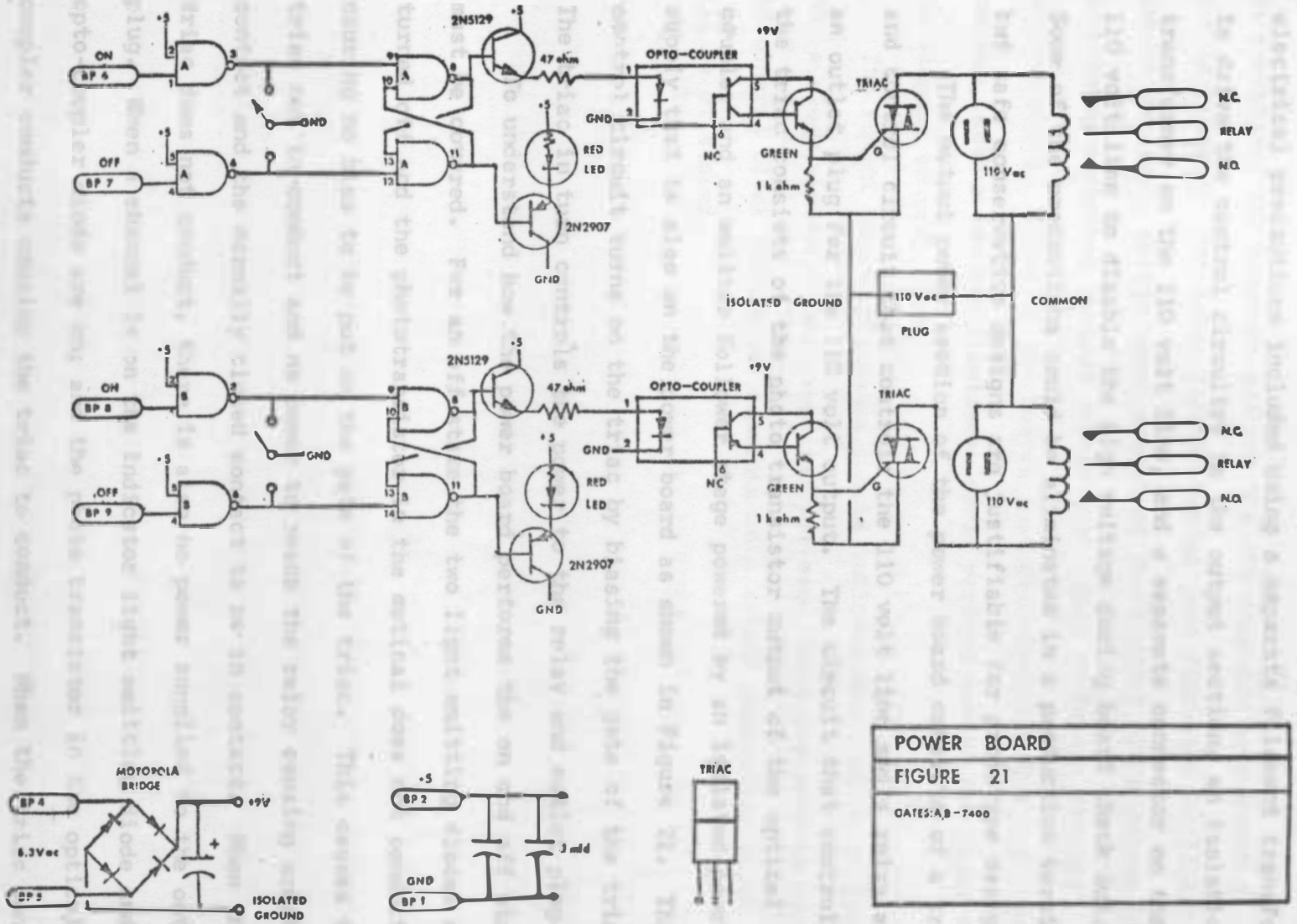
E. Power Board

Since many industrial applications require the control of lighting or other electrical equipment, it was only reasonable to design a module that could control these devices and be compatible to the digital clock system. It was decided to provide two types of output control: 110 volt a.c. for small standard equipment and a set of controllable relay contacts for controlling heavy equipment. This relay contact arrangement makes the digital clock system compatible with existing systems using older electromechanical control clocks.

The power board that was created has two separate control channels that are identical so the description will not be duplicated for both channels. Each channel can be divided into two sections, the digital logic section and the ouput power section, with the two sections separated by an optical coupler.

Each logic section consists of an S-R flip-flop that can be set or reset either manually or by the microcomputer as shown in Figure 21. The Q or positive flip-flop output controls the light emitting diode in the optical coupler and the Q or inverted flip-flop output is used to control LED indicator. It should be noted that when the flip-flop is in an on state both the optical coupler LED and the indicator LED are on and when the flip-flop is in an off state both LEDs are off.

The optical coupler, previously mentioned, functions to electrically separate the logic section from the output power sections. This is done to reduce the danger of electrical shock and logic circuit damage from the high voltage levels in the output section. Other



POWER BOARD
FIGURE 21
GATES: A,B - 7400

electrical precautions included using a separate filament transformer to drive the control circuitry in the output section, an isolation transformer on the 110 volt line, and a separate connector on the 110 volt line to disable the high voltage during board check out. Some of the precautions could be eliminated in a production version, but safe conservative designs are justifiable for prototype designs.

The output power section of the power board consists of a triac and control circuit that controls the 110 volt line and a relay and an outlet plug for the 110 volt output. The circuit that controls the triac consists of the photo transistor output of the optical coupler and an emitter follower stage powered by an isolated power supply that is also on the power board as shown in Figure 21. This control circuit turns on the triac by biasing the gate of the triac. The triac in turn controls the power to the relay and outlet plug.

To understand how the power board performs the on and off states must be covered. For an off state the two light emitting diodes are turned off and the phototransistor in the optical does not conduct, causing no bias to be put on the gate of the triac. This causes the triac not to conduct and no power to reach the relay causing armature contact and the normally closed contact to be in contact. When the triac does not conduct, there is also no power supplied to the outlet plug. When a channel is on the indicator light emitting diode and the opto-coupler diode are on; and the photo transistor in the optical coupler conducts causing the triac to conduct. When the triac conducts, power is supplied to both the relay and outlet plug. When the relay

is energized, the armature contact and the normally open contact are in contact. The relay contact system just discussed fulfilled one of the early design requirements of the digital clock system of developing a digital control system that could emulate a simplex control clock. Three of the boards just described and the main clock modules, with a few modifications, could perform easily the tasks of the simplex control clock.

The output boards discussed should indicate the variety of tasks that a clock system of this type can perform. Some assessments of the general clock system and comments on a second generation clock will be discussed.

CHAPTER IV

Summary

The digital clock system has been an interesting and informative project, exhibiting a design transition from random logic to a microprocessor based system. In the areas where random logic was used many different design approaches were incorporated. These ranged from the unique modifications made to existing circuits such as the incorporation of a calculator in the phase of the moon board and the solid state music box in the Westminster board to totally new circuits and approaches. One system that exhibited both original circuitry and approach was the calendar board. The algorithm that was developed was not the traditional leap year test of divisibility by four but rather a binary bit pattern test easily implemented in logic.

The microprocessor controlled decision section exhibited several important design features. First, it was shown that a successful system could be developed using only a small number of components. The input-output section illustrated how control flags could be used as output signals to minimize circuitry. The system software was successful in both system operation and system diagnostics. The microcomputer, while simple in nature, provided a means of learning the intricacies of a microprocessor based system.

Of equal importance to the system software and circuitry design was the modularity and physical layout of the system. The modularity

of the boards not only provided versatility in system configuration but facilitated diagnostics and fault location. Within each board components were located in groups related to the logical function they were to perform. This has the effect of further fault location and facilitated changes in the logic sections when needed.

When the design of a system is completed, a question that naturally arises is, "What would be done differently on the next design of the system?" These differences can be categorized into several major headings including remedial or changes to correct deficiencies in the performance or fabrication, changes in performance or enhanced features, and changes in internal structure.

Due to the relative success of the system, relatively few circuit changes are needed. The only major problem in the clock system, as designed, is a thermal problem due to the close packing of the circuitry. This could easily be corrected by spacing the boards farther apart, by using lower powered logic, or mounting boards in a vertical plane rather than in a horizontal plane.

Due to the modular nature of the clock system, numerous performance enhancements could be added to the system. Two features that were considered but were deferred to a second generation system were a radio receiver to synchronize the clock to the National Bureau of Standard's WWVB time standard and a board to transmit the time in an asynchronous serial code that could be used by slave clocks throughout a building.

If the decision was made to completely redesign the internal structure of the clock system, there would be a continuation of the transition from random logic to a microcomputer based system. Programs stored in memory would replace networks of random logic and much of the response circuitry would be reduced to interface circuitry to a computer input-output port. The computer section in turn would not only be handling the question, "Should anything be done at this time?" but rather actually handling much of the response sections. A clock comprised of a few interface circuits and a microcomputer with most functions implemented as programs could be designed and constructed in a much shorter time than the extensive random logic arrays used in this clock.

The design, construction, and documentation of the digital clock system has been informative and challenging. It is hoped that completing this project will be considered time well spent.

BIBLIOGRAPHY

Altman, Laurence, ed. Microprocessors. Electronics Book Series, New York: McGraw-Hill, 1975.

Applications Engineering Department. Semiconductor Power Circuits Handbook. Phoenix: Motorola Semiconductor Products Inc., 1968.

Baillie, G. H., C. Clutton, and C. A. Ilbert. Britten's Old Clocks and Watches and Their Makers. Seventh Ed. New York: Bonanza Books, 1956.

Carr, William N. and Jack P. Mize. MOS/LSI Design and Application. Texas Instruments Electronics Series, New York: McGraw-Hill Inc., 1972.

Clare, Christopher R. Designing Logic Systems Using State Machines. New York: McGraw-Hill, 1973.

Corliss, William R. Computers. Oak Ridge: U.S. Atomic Energy Commission, 1968.

Davidow, Bill. From CPU to Software. Santa Clara: Intel Corp., 1974.

The Editors of Electronics Magazine. Designer's Casebook. New York: McGraw-Hill, 1976.

The Engineering Staff of Texas Instruments. The Linear and Interface Circuits Data Book for Design Engineers, Dallas: Texas Instruments Inc., 1973.

The Engineering Staff of Texas Instruments. The Opto Electronics Data Book for Design Engineers. Second Ed. Dallas: Texas Instruments Inc., 1975.

The Engineering Staff of Texas Instruments. Supplement to the TTL Data Book for Design Engineers. Dallas: Texas Instruments Inc., 1974.

The Engineering Staff of Texas Instruments. The TTL Data Book for Design Engineers. Dallas: Texas Instruments, 1973.

Helmers, Carl T. The Experimenter's Computer System ECS-3, Belmont: M. P. Publishing Co., 1974.

Hill, Bredrick J. and Gerald R. Peterson. Introduction to Switching Theory and Logical Design, 2nd Ed. New York: John Wiley and Sons, 1974.

Hoenig, Stuart A., F. Leland Payne. How to Build and Use Electronic Devices Without Frustration, Panic, Mountains of Money, or an Engineering Degree. Boston: Little Brown, 1973.

Hoff, Marcian, ed. The Intel Memory Design Handbook, Santa Clara: Intel Corp., 1973.

Hoyle, Fred. Astronomy. London: Rathbone Books Limited, 1962.

Intel Data Catalog 1976. Santa Clara: Intel Corp., 1976.

Intel 8008 Assembly Language Programming Manual. Santa Clara: Intel Corp., 1975.

Intel 8008 Users Manual. Second Printing. Santa Clara: Intel Corporation, 1973.

Intel 8080 Microcomputer Systems User's Manual. Santa Clara: Intel Corporation, 1975.

Intel MCS-8 User's Manual. Santa Clara: Intel Corporation, 1975.

Lancaster, Don. TTL Cookbook. Indianapolis: Howard W. Sams and Co. Inc., 1974.

Larsen, David G., Peter R. Rony, and Jonathan A. Titus. The Bug Book III Micro Computer Interfacing. Derby, Connecticut: E and L Instruments, Inc., 1975.

Luecke, Gerald, Jack P. Mize and William N. Carr. Semiconductor Memory Design and Application. Texas Instruments Electronics Series, New York: McGraw-Hill Inc., 1973.

Lynch, John. The Coffee Table Book of Astrology. New York: Viking Press, 1967.

Martin, Donald P, Microcomputer Design, 2nd Ed. Northbrook, Ill.: Martin Research, 1976.

McMos Integrated Circuits Data Book. Phoenix: Motorola, Inc., 1973.

MD 8008 Applications Manual. Ottawa: Micro Systems International, 1974.

MOS Integrated Circuits. Santa Clara: National Semiconductor Corp., 1974.

Myers, Robert ed. The Radio Amateur's Handbook. Fifty-third Ed. Newington: American Radio Relay League, 1976.

National Engineering Consortium. Micrccomputers - Principles and Applications. Chicago: National Engineering Consortium, 1975.

Osborne and Associates. An Introduction to Microcomputers. Berkeley: Osborne and Associates, 1975.

Pippenger, D. E. and C. L. McCollum. Linear and Interface Circuits Applications. Dallas: Texas Instruments Inc., 1974.

Reti, Ladislal ed. The Unknown Leonardo. New York: McGraw-Hill, 1974.

Richman, Paul. MOS Field-Effect Transistors and Integrated Circuits. New York: John Wiley and Sons, 1973.

Semiconductor Data Library. Five Vols. Phoenix: Motorola Semiconductor Products Inc., 1974.

Sevastopoulos, Nello, et. al. National Voltage Regulator Handbook. Santa Clara: National Semiconductor Corp., 1975.

Signetics: Digital, Linear, MOS Applications. Sunnyvale, California: Signetics Corp., 1974.

Signetics: Digital, Linear, MOS Data Book. Sunnyvale, California: Signetics Corp., 1974.

Titus, Jonathan A. Computer! New York: Gernsback Publications Inc., 1974.

Torrero, Edward A. ed. Microprocessors New Directions for Designers. Rochelle Park: Hayden Book Co., Inc., 1975.

The Vada Company. Guide to Standard MOS Products. Santa Clara: American Microsystems Inc., 1975.

Vollmer, Emil ed. Leonardo Da Vinci. New York: Renal and Co., 1956

Weitzman, Cay. Minicomputer Systems. Englewood Cliffs: Prentice-Hall Inc., 1974.

Wester, John G. and William D. Simpson. Software Design for Microprocessors. Dallas: Texas Instruments Learning Center, 1976.

Westman, H. P. et al. Reference Data for Radio Engineers. Fifth Ed. Indianapolis: Howard W. Sams and Co. inc., 1974.

Wickes, William E. Logic Design with Integrated Circuits. New York: John Wiley and Sons, Inc., 1968.

Zener Diode Handbook. Phoenix: Motorola Semiconductor Products Inc., 1967.

Footnotes

1. J. R. R. Tolkien, The Hobbit (New York: Ballantine Books, 1966), pp. 84-85.

2. G. H. Baillie, Britten's Old Clocks and Watches and Their Makers (New York: Bonanza Books, 1956), p. 3.

3. G. H. Baillie, Britten's Old Clocks and Watches and Their Makers (New York: Bonanza Books, 1956), p. 67

4. Emil M. Buhrer, ed., The Unknown Leonardo (New York: McGraw-Hill, 1974), pp. 242-247.

5. David Eugene Smith, "Abacus" Encyclopaedia Britannica (Chicago: Encyclopaedia Britannica Inc., 1970), Vol. I, p. 6.

6. William R. Corliss, Computers (Oak Ridge: U.S. Atomic Energy Commission, 1968), p.4.

7. William R. Corliss, Computers (Oak Ridge: U.S. Atomic Energy Commission, 1968), p. 4.

8. William R. Corliss, Computers (Oak Ridge: U.S. Atomic Energy Commission, 1968), p. 9.

9. William R. Corliss, Computers (Oak Ridge: U.S. Atomic Energy Commission, 1968), pp. 10-11.

10. F. Faggin, M. E. Hoff, "Standard Parts and Custom Design Merge in Four-Chip Processor Kit," Electronics April 24, 1972, pp. 112-116.

11. Adam Osborne, An Introduction to Microcomputers (Berkeley: Osborne and Associates, 1975), pp. 1-4.

12. Adam Osborne, An Introduction to Microcomputers (Berkeley: Osborne and Associates, 1975), pp. 1-4.

13. Simplex Time Recorder Company, Simplex Program Machine (Gardner, Massachusetts: Simplex Time Recorder Company), pp. 1-2.

14. Gary L. Wilhelm, A Special Purpose Computer and Console to Assist Blind Bowlers (Unpublished Master's Dissertation, Dept. of Electrical Engineering, South Dakota State University, 1972), pp. 1-56.

15. William R. Corliss, Computers (Oak Ridge: U.S. Atomic Energy Commission, 1968), p. 5.

16. Don Lancaster, TTL Cookbook (Indianapolis: Howard W. Sams and Company 1976), p. 231.

17. MOS Integrated Circuits (Santa Clara: National Semiconductor Corp., 1974), pp. 10-14

18. M. V. Wilkes, "The Best Way To Design An Automatic Calculating Machine" Report of the Manchester University Computer Inaugural Conference (Manchester, England: Manchester University, 1951), pp. 16-18.

19. Donald P. Martin, Microcomputer Design (Northbrook, Ill.: Martin Research, 1976), pp. 25-3 - 25-5.

20. MCS-8 Microcomputer Set Users Manual (Santa Clara: Intel Corp., 1974), pp. 1-35.

21. Donald P. Martin, Microcomputer Design (Northbrook, Ill.: Martin Research, 1976), p. 24-4.

22. Intel 8008 Assembly Language Programming Manual (Santa Clara: Intel Corp., 1975), pp. 1-52.

23. Don Frohman-Bentchkowsky, "A Fully Decoded 2048-Bit Electrically-Programmable MOS Read Only Memory" ISSCC Digest of Technical Papers, February 1971, pp. 80-82.

24. Nello Sevastopoulos, et al National Voltage Regulator Handbook (Santa Clara: National Semiconductor Corporation, 1975), pp. 7-1 - 8-10.

25. GEL/CELL Rechargeable Batteries Charging Manuals (Milwaukee: Globe-Union Inc.), p. 4.

26. Clocks and Clockworks (Elmhurst, Ill: Craft Products, 1969), p. 16.

27. H. Alan Lloyd, The Collector's Dictionary of Clocks (New York: A. S. Barnes and Co., 1964), p. 39.

28. H. Alan Lloyd, The Collector's Dictionary of Clocks (New York: A. S. Barnes and Co., 1964), p. 40.

29. Wu Chih-Ch'ang, "Calendar" Encyclopaedia Britannica (Chicago: Encyclopaedia Britannica Inc., 1970), Vol. IV, pp. 611-612.

30. Wu Shih-Ch'ang "Calendar" Encyclopaedia Britannica (Chicago: Encyclopaedia Britannica Inc., 1970), Vol. IV, pp. 615-616.

31. Wu Chih-Ch'and "Calendar" Encyclopaedia Britannica (Chicago: Encyclopaedia Britannica Inc., 1970), Vol. IV, pp. 617-619.

32. Wu Chih-Ch'and "Calendar" Encyclopaedia Britannica (Chicago: Encyclopaedia Britannica Inc., 1970), Vol. IV, p. 614.

33. Joseph Proudman and Gordon William Groves "Tide" Encyclopaedia Britannica (Chicago: Encyclopaedia Britannica Inc., 1970) Vol. XXI, p. 1126.

34. MOS Integrated Circuits (Santa Clara: National Semiconductor Corp., 1974), pp. 10-54 - 10-58.

APPENDIX A

CLOCK SYSTEM SOFTWARE

1702 READ ONLY MEMORY FOR CENTRAL PROCESSING BOARD

	address	instruct.	mnemonic	comment
1.	000	006	MVI A	
2.	001	000	000	loads zeros into accum.
3.	002	006	MVI A	
4.	003	000	000	
5.	004	310	MOV B,A	
6.	005	320	MOV C,A	
7.	006	330	MOV D,A	
8.	007	340	MOV E,A	
9.	010	350	MOV H,A	
10.	011	360	MOV L,A	
11.	012	103	IN PORT 1	Point A
12.	013	273	CMP D	Compare if time has changed.
13.	014	150	JZ	Jump back if zero.
14.	015	012	(to point A)	
15.	016	000		
16.	017	103	In Port 1	Read again to see if still different.
17.	020	273	CMP D	
18.	021	150	JZ	If 0 then it was just a noise glitch.
19.	022	012	(to Point A)	If not equal, then go on.
20.	023	000		
21.	024	330	MOV D,A	Point B Update least significant Byte.
22.	025	103	In Port 1	
23.	026	273	CMP D	
24.	027	110	JNZ	Repeat READ if not equal.
25.	030	024	(to Point B)	
26.	031	000		
27.	032	101	In Port 0	
28.	033	320	MOV C,A	Point C
29.	034	101	In Port 0	Get debounced hours.
30.	035	272	CMP C	
31.	036	110	JNZ	
32.	037	033	(to Point C)	
33.	040	000		
34.	041	303	MOV A,D	

1702 READ ONLY MEMORY FOR CENTRAL PROCESSING BOARD

	address	instruct.	mnemonic	comment
35.	042	044	ANI	
36.	043	017		New
37.	044	110	JNZ	
38.	045	050		Skip in Port 2 if not equal to 10x.
39.	046	000		
40.	047	105	In Port 2	
41.	050	056	MVI H	Clear H if not already clear.
42.	051	000	000	
43.	052	066	MVI L	This address is the start of command table.
44.	053	117		(3 Bytes less)
45.	054	060	INR L	Point D
46.	055	060	INR L	Point E
47.	056	060	INR L	Point F
48.	057	306	MOV A,L	
49.	060	074	CPI	This checks for end of command table.
50.	061	372		
51.	062	100	JNC	
52.	063	012	(to Point A)	
53.	064	000		
54.	065	317	MOV B,M	Most significant Byte of command.
55.	066	301	MOV A,B	Table to Reg. B
56.	067	044	ANI	Mask off everything but week day/week end bits.
57.	070	300		
58.	071	242	ANA C	And days of command table w/real time days.
59.	072	150	JZ	
60.	073	054	(to Point D)	If days don't match go to next set in command table.
61.	074	000		
62.	075	301	MOV A,B	
63.	076	252	XRA C	This compares most sig.
64.	077	044	ANI	Byte without day's aection.
65.	100	077		
66.	101	110	JNZ	
67.	102	054	(to Point D)	
68.	103	000		

1702 READ ONLY MEMORY FOR CENTRAL PROCESSING BOARD

	address	instruct.	mnemonic	comment
69.	104	060	INR L	At this point first Bytes match increment to next Byte in table.
70.	105	307	MOV A,M	
71.	106	273	CMP D	Compare least sig. Byte.
72.	107	110	JNZ	Skip back and increment if no match.
73.	110	055	(to Point E)	
74.	111	000		
75.	112	060	INR L	
76.	113	307	MOV A,M	
77.	114	121	Out Port 10	Output command Byte
78.	115	104	JMP	
79.	116	056	(to Point F)	
80.	117	000		
81.	120	000	HLT	
82.	121	000	HLT	
83.	122	111		9:00 a.m. Command table starts here. Weekday only
84.	123	200		
85.	124	100	Westminster	
86.	125	111		9:15 a.m. Weekday only Westminster
87.	126	225		
88.	127	100		9:30
89.	130	111		
90.	131	260		
91.	132	100		9:45
92.	133	111		
93.	134	305		
94.	135	100		10:00
95.	136	320		
96.	137	200		Set channel 1 Power Board.
97.	140	110		10:15
98.	141	320		
99.	142	225		Reset channel 1 Power Board.
100.	143	104		

1702 READ ONLY MEMORY FOR CENTRAL PROCESSING BOARD

	address	instruct.	mnemonic	comment
101	144	320		10:30
102	145	260		
103	146	100		
104	147	320		10:45
105	150	305		
106	151	100		
107	152	321		11:00
108	153	200		
109	154	100		
110	155	321		11:15
111	156	225		
112	157	100		
113	160	321		11:30
114	161	260		
115	162	100		
116	163	321		11:45
117	164	305		
118	165	100		
119	166	322		12:00 noon
120	167	200		
121	170	100		
122	171	322		12:15 p.m.
123	172	225		
124	173	140		Set intrusion
125	174	322		12:30
126	175	260		
127	176	100		
128	177	322		12:45
129	200	305		
130	201	100		
131	202	323		13:00
132	203	200		
133	204	102		Set Channel 2
134	205	323		13:15
135	206	225		
136	207	101		

1702 READ ONLY MEMORY FOR CENTRAL PROCESSING BOARD

	address	instruct.	mnemonic	comment
137	210	323		13:30
138	211	260		
139	212	100		
140	213	323		13:45
141	214	305		
142	215	100		
143	216	324		14:00
144	217	200		
145	220	110		Set Channel 1
146	221	324		14:15
147	222	225		
148	223	104		Reset Channel 1
149	224	324		14:30
150	225	260		
151	226	100		
152	227	324		14:45
153	230	305		
154	231	100		
155	232	325		15:00
156	233	200		
157	234	100		
158	235	325		15:15
159	236	225		
160	237	100		
161	240	325		15:30
162	241	260		
163	242	100		
164	243	325		15:45
165	244	305		
166	245	100		
167	246	326		16:00
168	247	200		
169	250	100		
170	251	326		16:15
171	252	225		
172	253	100		
173	254	326		16:30
174	255	260		
175	256	100		
176	257	326		16:45
177	260	305		
178	261	100		

1702 READ ONLY MEMORY FOR CENTRAL PROCESSING BOARD

	address	instruct.	mnemonic	comment
179	262	327		17:00
180	263	200		
181	264	100		
182	265	327		17:15
183	266	225		
184	267	100		
185	270	327		17:30
186	271	260		
187	272	100		
188	273	327		17:45
189	274	305		
190	275	140		West and Intrusion set.
191	276	330		18:00
192	277	200		
193	300	100		
194	301	330		18:15
195	302	225		
196	303	100		
197	304	330		18:30
198	305	260		
199	306	100		
200	307	330		18:45
201	310	305		
202	311	100		
203	312	331		19:00
204	313	200		
205	314	100		
206	315	331		19:15
207	316	225		
208	317	100		
209	320	331		19:30
210	321	260		
211	322	100		
212	323	331		19:45
213	324	305		
214	325	100		
215	326	340		20:00
216	327	200		
217	330	100		

1702 READ ONLY MEMORY FOR CENTRAL PROCESSING BOARD

	address	instruct.	mnemonic	comment
218	331	340		20:15
219	332	225		
220	333	100		
221	334	340		20:30
222	335	260		
223	336	100		
224	337	340		20:45
225	340	305		
226	341	100		
227	342	341		21:00
228	343	200		
229	344	100		
230	345	241		21:15
231	346	225		
232	347	100		
233	350	241		21:30
234	351	260		
235	352	100		
236	353	241		21:45
237	354	305		
238	355	100		
239	356	242		22:00
240	357	200		
241	360	100		
242	361	300		00:00 Day Strobe
243	362	200		
244	363	200		
245	364	107		7:45 Reset Intrusion m-F
246	365	305		
247	366	020		
248	367	122		12:50 Reset Intrusion m-F
249	370	320		
250	371	020		
251	372	000		
252	373	000		
253	374	000		
254	375	000		
255	376	000		
256	377	000		

1702 READ ONLY MEMORY FOR CENTRAL PROCESSING BOARD
(DIAGNOSTIC PROGRAMS)

	address	instruct.	mnemonic	comment
1.	000	103	In Port 1	Input Port 1
2.	001	121	Out Port 10	Output Port 8
3.	002	104	JMP	Jump back
4.	003	000		Address
5.	004	000		

1.	000	101	In Port 0	Input Port 0
2.	001	121	Out Port 10	Output Port 8
3.	002	104	JMP	Jump back
4.	003	000		Address
5.	004	000		

APPENDIX B

CODE TABLE AND BUS ORGANIZATION LISTS

BACK PLANE INTERCONNECT LIST

Display Board

Pin 1.	Ground	7.	Day code-A
2.	Plus 4.3 volts	8.	PM low
3.	Plus side of ground disable	9.	AM low
4.	Day code-D	10.	Power out low
5.	Day code-C	11.	Negative ground disable
6.	Day code-B		

Overflow Board

Pin 1.	Ground	7.	Day strobe input
2.	Plus five volts	8.	Activate display
3.	Minus ten volts	9.	Activate display
4.	60 Hz. from transformer	10.	Weekend flag
5.	60 Hz. from transformer	11.	Outage status
6.	60 Hz to timing board		

Timing Board

Pin 1.	Ground	7.	Day
2.	Plus five volts	8.	60 Hz.
3.	Minus ten volts	9.	10 minute update
4.	Day	10.	AM low
5.	Day	11.	PM low
6.	Day		

Central Processing Unit Board

Pin 1.	Ground	7.	Reset intrusion
2.	Plus five volts	8.	Set channel one
3.	Minus ten volts	9.	Reset channel one
4.	Day strobe	10.	Set channel two
5.	Westminster strobe	11.	Reset channel two

Westminster Board

Pin 1.	Ground	7.	No connection
2.	Plus five volt	8.	No connection
3.	Start strobe	9.	No connection
4.	6.3 volt A.C.	10.	No connection
5.	6.3 volt A.C.	11.	No connection
6.	No connection		

BACK PLANE INTERCONNECT LIST

Intrusion Board

Pin 1.	Ground	7.	No connection
2.	Plus five volts	8.	No connection
3.	No connection	9.	No connection
4.	Set intrusion	10.	No connection
5.	Reset intrusion	11.	No connection
6.	No connection		

Calendar Board

Pin 1.	Ground	7.	Zodiac - B
2.	Plus five volts	8.	Zodiac - A
3.	Minus ten volts	9.	No connection
4.	Day strobe	10.	No connection
5.	Zodiac - D	11.	Unprotected plus five volts
6.	Zodiac - C		

Phase of the Moon Board

Pin 1.	Ground	7.	Zodiac Bit - 2
2.	Plus five volts	8.	Zodiac Bit - 1
3.	Minus ten volts	9.	No connection
4.	10 minute update	10.	No connection
5.	Zodiac Bit - 8	11.	No connection
6.	Zodiac Bit - 4		

Power Board

Pin 1.	Ground	7.	Reset channel one
2.	Plus five volts	8.	Set channel two
3.	No connection	9.	Reset channel two
4.	6-3 volt A.C.	10.	No connection
5.	6-3 volt A.C.	11.	No connection
6.	Set channel one		

Battery Board

Pin 1.	No connection	7.	No connection
2.	Plus voltage	8.	No connection
3.	No connection	9.	No connection
4.	No connection	10.	No connection
5.	No connection	11.	Minus voltage
6.	No connection		

DATA BUS ORGANIZATION FOR DISPLAY BOARD

DISPLAY CABLES

Hour-minute cable

Line 1.	Minute Bit-1	Line 9.	Hour Bit-1
2.	Minute Bit-2	10.	Hour Bit-2
3.	Minute Bit-4	11.	Hour Bit-4
4.	Minute Bit-8	12.	Hour Bit-8
5.	10 minute Bit-1	13.	10 hour Bit-1
6.	10 minute Bit-2	14.	10 hour Bit-2
7.	10 minute Bit-4	15.	No connection
8.	No connection	16.	No connection

Second cable

Line 1.	Second Bit-1	5.	10 second Bit-1
2.	Second Bit-2	6.	10 second Bit-2
3.	Second Bit-4	7.	10 second Bit-4
4.	Second Bit-8	8.	10 second Bit-8

Day Cable

Line 1.	QD see day code table
2.	QC
3.	QB
4.	QA

CODE FORMAT FOR DAY CODE

Day	Number	Qd	Qc	Qb	Qa
Tuesday	0	0	0	0	0
Wednesday	1	0	0	0	1
Thursday	2	0	0	1	0
Friday	3	0	0	1	1
Saturday	4	0	1	0	0
Sunday	5	0	1	0	1
Monday	9	1	0	0	1

Qc = weekend flag

DATA BUS ORGANIZATION FOR TWELVE AND TWENTY FOUR HOUR DISPLAY

Bit 0. 12 hour mode BCD code of hour - 2
1. 12 hour mode BCD code of hour - 4
2. 12 hour mode BCD code of hour - 8
3. 12 hour mode BCD code of 10 hour - 1
4. 12 hour mode BCD code of 10 hour - 2
5. AM = low
6. Lunch light
7. Coffee break light

8223 READ ONLY MEMORY FOR TWELVE AND TWENTY FOUR HOUR DISPLAY

	address	data	comment	
1.	000	300	00 - 00	(24 hour mode - 12 hour mode)
2.	001	300	01 - 01	
3.	002	301	02 - 02	
4.	003	301	03 - 03	
5.	004	302	04 - 04	
6.	005	302	05 - 05	
7.	006	303	06 - 06	
8.	007	303	07 - 07	
9.	010	304	08 - 08	
10.	011	304	09 - 09	
11.	012	110	10 - 10	
12.	013	310	11 - 11	
13.	014	251	12 - 12	
14.	015	340	13 - 01	
15.	016	141	14 - 02	
16.	017	341	15 - 03	
17.	020	142	16 - 04	
18.	021	342	17 - 05	
19.	022	343	18 - 06	
20.	023	343	19 - 07	
21.	024	344	20 - 08	
22.	025	344	21 - 09	
23.	026	350	22 - 10	
24.	027	350	23 - 11	
25.	030	x		(x = not used)
26.	031	x		
27.	032	x		
28.	033	x		
29.	034	x		
30.	035	x		
31.	036	x		
32	037	x		

DATA BUS ORGANIZATION FOR CENTRAL PROCESSING BOARD

Ribbon Cable for Real Time Input to CPU Board

Line 1.	Week end	9.	10 minute strobe
2.	Week day	10.	10 minute Bit-4
3.	10 hour Bit-2	11.	10 minute Bit-2
4.	10 hour Bit-1	12.	10 minute Bit-1
5.	Hour Bit-8	13.	Minute Bit-8
6.	Hour Bit-4	14.	Minute Bit-4
7.	Hour Bit-2	15.	Minute Bit-2
8.	Hour Bit-1	16.	Minute Bit-1

READ ONLY MEMORY DATA BUS CODE ORGANIZATION

CPU COMMAND TABLE

First Byte		Second Byte	
Bit 0.	Hour - 1	Bit 0.	Minute 1
1.	Hour - 2	1.	Minute 2
2.	Hour - 4	2.	Minute 4
3.	Hour - 8	3.	Minute 8
4.	10 hour - 1	4.	10 minute 1
5.	10 hour - 2	5.	10 minute 2
6.	Week Day	6.	10 minute 4
7.	Week End	7.	Not used

Third Byte

Bit 0.	Reset channel 2
1.	Set channel 2
2.	Reset channel 1
3.	Set channel 1
4.	Reset intrusion
5.	Set intrusion
6.	Westminster
7.	Day strobe

DATA BUS ORGANIZATION FOR WESTMINSTER BOARD

WESTMINSTER RIBBON CABLES

Time Cable

Line 1.	Minute Bit-4	5.	Hour Bit-8
2.	Minute Bit-2	6.	Hour Bit-4
3.	Minute Bit-1	7.	Hour Bit-2
4.	10 hour Bit-1	8.	Hour Bit-1

Board Interconnect Cable

Line 1.	6.3 volt AC	9.	ROM address A2
2.	6.3 volt AC	10.	ROM address A3
3.	8 OHM speaker	11.	ROM address A4
4.	8 OHM speaker (GND)	12.	To 74123 (mute)
5.	Low start	13.	From 7420 (mute)
6.	High start	14.	Ground
7.	ROM address AO	15.	No connection
8.	ROM address A1	16.	No connection

DATA BUS ORGANIZATION FOR 8223 WESTMINSTER CONTROL READ ONLY MEMORY

Bit 0. Quarter hour strike
1. Half hour strike
2. Three quarter strike
3. Hour strike

Bit 4-7 not used

BINARY CODING FOR MUSICAL SCALE COVERING TWO OCTAVES

	Musical Note	B5	B4	B3	B2	B1	B0
	E5	0	0	0	0	0	1
	E5	0	0	0	0	1	1
	D5	0	0	0	1	0	1
	C5#	0	0	0	1	1	1
	C5	0	0	1	0	0	1
	B4	0	0	1	1	0	0
	B4	0	0	1	1	1	0
	A4	0	1	0	0	0	1
	G4#	0	1	0	1	0	0
	G4	0	1	0	1	1	1
	F4#	0	1	1	0	1	1
	F4	0	1	1	1	1	0
	E4	1	0	0	0	0	1
	D4#	1	0	0	0	1	1
	D4	1	0	0	1	0	1
	C4#	1	0	0	1	1	1
Middle "c"	C4	1	0	1	0	0	1
	B3	1	0	1	1	0	0
	B3	1	0	1	1	1	0
	A3	1	1	0	0	0	1
	G3#	1	1	0	1	0	0
	G3	1	1	0	1	1	1
	F3#	1	1	1	0	1	1
	F3	1	1	1	1	1	0
	MUTE	0	0	1	1	1	1

BINARY CODE FOR PROGRAMMING BEATS PER NOTE

Beats Per Note	B7	B6
1	0	0
2	0	1
3	1	0
4	1	1

8223 READ ONLY MEMORY FOR WESTMINSTER MELODY

	address	data	comment	hour strike
1.	000	041	E	076
2.	001	045	D	017
3.	002	051	C	076
4.	003	167	G	017
5.	004	017	mute	076
6.	005	051	C	017
7.	006	041	E	076
8.	007	045	D	017
9.	010	167	G	076
10.	011	017	mute	017
11.	012	051	C	076
12.	013	045	D	017
13.	014	041	E	076
14.	015	151	C	017
15.	016	017	mute	076
16.	017	041	E	017
17.	020	051	C	076
18.	021	045	D	017
19.	022	167	G	076
20.	023	017	mute	017
21.	024	067	G	076
22.	025	045	D	017
23.	026	041	E	076
24.	027	151	C	017
25.	030	017	mute	017
26.	031	041	E	017
27.	032	045	D	017
28.	033	051	C	017
29.	034	167	G	017
30.	035	017	mute	017
31.	036	017	mute	017
32.	037	017	mute	017

8223 READ ONLY MEMORY FOR WESTMINSTER CONTROL CIRCUIT

	address	data	comment	
1.	000	001	0001	
2.	001	001	0001	
3.	002	001	0001	$\frac{1}{4}$ hour
4.	003	001	0001	
5.	004	001	0001	
6.	005	012	1010	
7.	006	012	1010	
8.	007	012	1010	
9.	010	012	1010	
10.	011	012	1010	$\frac{1}{2}$ hour
11.	012	012	1010	
12.	013	012	1010	
13.	014	012	1010	
14.	015	012	1010	
15.	016	012	1010	
16.	017	014	1100	
17.	020	014	1100	
18.	021	014	1100	
19.	022	014	1100	
20.	023	014	1100	
21.	024	014	1100	3/4 hour
22.	025	014	1100	
23.	026	014	1100	
24.	027	014	1100	
25.	030	014	1100	
26.	031	004	0100	
27.	032	004	0100	
28.	033	004	0100	
29.	034	004	0100	
30.	035	004	0100	
31.	036	000	0000	
32.	037	000	0000	

READ ONLY MEMORY DATA BUS ORGANIZATION

Calendar Look Up Table

Least Significant Byte (month and day)

Bit 1. Day Bit-1
2. Day Bit-2
3. Day Bit-4
4. Day Bit-8
5. 10 Day Bit-1
6. 10 Day Bit-2
7. Month Bit-1
8. Month Bit-2

Most Significant Byte (Zodiac and month)

Bit 1. Month Bit-4
2. Month Bit-8
3. 10 Month Bit-1
4. Reset Bit 1 = Feb. 29 or end of year
5. Zodiac Bit-1
6. Zodiac Bit-2
7. Zodiac Bit-4
8. Zodiac Bit-8

1702 READ ONLY MEMORY LOOK UP TABLE FOR CALENDAR BOARD

	address	least significant Byte data	comments		most significant Byte data
1.	000	101	Jan.	1	240
2.	001	102		2	240
3.	002	103		3	240
4.	003	104		4	240
5.	004	105		5	240
6.	005	106		6	240
7.	006	107		7	240
8.	007	110		8	240
9.	010	111		9	240
10.	011	120		10	240
11.	012	121		11	240
12.	013	122		12	240
13.	014	123		13	240
14.	015	124		14	240
15.	016	125		15	240
16.	017	126		16	240
17.	020	127		17	240
18.	021	130		18	240
19.	022	131		19	240
20.	023	140		20 Aquarius	260
21.	024	141		21	260
22.	025	142		22	260
23.	026	143		23	260
24.	027	144		24	260
25.	030	145		25	260
26.	031	146		26	260
27.	032	147		27	260
28.	033	150		28	260
29.	034	151		29	260
30.	035	160		30	260
31.	036	161		31	260
32.	037	201	Feb.	1	260
33.	040	202		2	260
34.	041	203		3	260
35.	042	204		4	260
36.	043	205		5	260
37.	044	206		6	260

1702 READ ONLY MEMORY LOOK UP TABLE FOR CALENDAR BOARD

	address	least significant Byte data	comments	most significant Byte data
38.	045	207	Feb. 7	260
39.	046	210	8	260
40.	047	211	9	260
41.	050	220	10	260
42.	051	221	11	260
43.	052	222	12	260
44.	053	223	13	260
45.	054	224	14	260
46.	055	225	15	260
47.	056	226	16	260
48.	057	227	17	260
49.	060	230	18	260
50.	061	231	19 Pisces	300
51.	062	240	20	300
52.	063	241	21	300
53.	064	242	22	300
54.	065	243	23	300
55.	066	244	24	300
56.	067	245	25	300
57.	070	246	26	300
58.	071	247	27	300
59.	072	250	28	300
60.	073	251	29, reset	310
61.	074	301	Mar. 1	300
62.	075	302	2	300
63.	076	303	3	300
64.	077	304	4	300
65.	100	305	5	300
66.	101	306	6	300
67.	102	307	7	300
68.	103	310	8	300
69.	104	311	9	300
70.	105	320	10	300
71.	106	321	11	300
72.	107	322	12	300
73.	110	323	13	300

1702 READ ONLY MEMORY LOOK UP TABLE FOR CALENDAR BOARD

	address	least significant Byte data	comments		Byte significant Byte data
74.	111	324	Mar. 14		300
75.	112	325	15		300
76.	113	326	16		300
77.	114	327	17		300
78.	115	330	18		300
79.	116	331	19		300
80.	117	340	20		300
81.	120	341	21	Aries	020
82.	121	342	22		020
83.	122	343	23		020
84.	123	344	24		020
85.	124	345	25		020
86.	125	346	26		020
87.	126	347	27		020
88.	127	350	28		020
89.	130	351	29		020
90.	131	360	30		020
91.	132	361	31		020
92.	133	001	Apr. 1		021
93.	134	002	2		021
94.	135	003	3		021
95.	136	004	4		021
96.	137	005	5		021
97.	140	006	6		021
98.	141	007	7		021
99.	142	010	8		021
100.	143	011	9		021
101.	144	020	10		021
102.	145	021	11		021
103.	146	022	12		021
104.	147	023	13		021
105.	150	024	14		021
106.	151	025	15		021
107.	152	026	16		021
108.	153	027	17		021
109.	154	030	18		021

1702 READ ONLY MEMORY LOOK UP TABLE FOR CALENDAR BOARD

	address	least significant Byte data	comments		most significant Byte data
110.	155	031	Apr.	19	021
111.	156	040		20 Taurus	041
112.	157	041		21	041
113.	160	042		22	041
114.	161	043		23	041
115.	162	044		24	041
116.	163	045		25	041
117.	164	046		26	041
118.	165	047		27	041
119.	166	050		28	041
120.	167	051		29	041
121.	170	060		30	041
122.	171	101	May	1	041
123.	172	102		2	041
124.	173	103		3	041
125.	174	104		4	041
126.	175	105		5	041
127.	176	106		6	041
128.	177	107		7	041
129.	200	110		8	041
130.	201	111		9	041
131.	202	120		10	041
132.	203	121		11	041
133.	204	122		12	041
134.	205	123		13	041
135.	206	124		14	041
136.	207	125		15	041
137.	210	126		16	041
138.	211	127		17	041
139.	212	130		18	041
140.	213	131		19	041
141.	214	140		20	041
142.	215	141		21 Gemini	061
143.	216	142		22	061
144.	217	143		23	061
145.	220	144		24	061

1702 READ ONLY MEMORY LOCK UP TABLE FOR CALENDAR BOARD

	address	least significant Byte data	comments	most significant Byte data
146.	221	145	May 25	061
147.	222	146	26	061
148.	223	147	27	061
149.	224	150	28	061
150.	225	151	29	061
151.	226	160	30	061
152.	227	161	31	061
153.	230	201	June 1	061
154.	231	202	2	061
155.	232	203	3	061
156.	233	204	4	061
157.	234	205	5	061
158.	235	206	6	061
159.	236	207	7	061
160.	237	210	8	061
161.	240	211	9	061
162.	241	220	10	061
163.	242	221	11	061
164.	243	222	12	061
165.	244	223	13	061
166.	245	224	14	061
167.	246	225	15	061
168.	247	226	16	061
169.	250	227	17	061
170.	251	230	18	061
171.	252	231	19	061
172.	253	240	20	061
173.	254	241	21	061
174.	255	242	22 Cancer	101
175.	256	243	23	101
176.	257	244	24	101
177.	260	245	25	101
178.	261	246	26	101
179.	262	247	27	101
180.	263	250	28	101
181.	264	251	29	101

1702 READ ONLY MEMORY LOOK UP TABLE FOR CALENDAR BOARD

	address	least significant Byte data	comments	most significant Byte data
182.	265	260	June 30	101
183.	266	301	July 1	101
184.	267	302	2	101
185.	270	303	3	101
186.	271	304	4	101
187.	272	305	5	101
188.	273	306	6	101
189.	274	307	7	101
190.	275	310	8	101
191.	276	311	9	101
192.	277	320	10	101
193.	300	321	11	101
194.	301	322	12	101
195.	302	323	13	101
196.	303	324	14	101
197.	304	325	15	101
198.	305	326	16	101
199.	306	327	17	101
200.	307	330	18	101
201.	310	331	19	101
202.	311	340	20	101
203.	312	341	21	101
204.	313	342	22	101
205.	314	343	23 Leo	121
206.	315	344	24	121
207.	316	345	25	121
208.	317	346	26	121
209.	320	347	27	121
210.	321	350	28	121
211.	322	351	29	121
212.	323	360	30	121
213.	324	361	31	121
214.	325	001	Aug. 1	122
215	326	002	2	122
216.	327	003	3	122
217.	330	004	4	122

1702 READ ONLY MEMORY LOOK UP TABLE FOR CALENDAR BOARD

	address	least significant Byte data	comments			most significant Byte data
218.	331	005	Aug.	5		122
219.	332	006		6		122
220.	333	007		7		122
221.	334	010		8		122
222.	335	011		9		122
223.	336	020		10		122
224.	337	021		11		122
225.	340	022		12		122
226.	341	023		13		122
227.	342	024		14		122
228.	343	025		15		122
229.	344	026		16		122
230.	345	027		17		122
231.	346	030		18		122
232.	347	031		19		122
233.	350	040		20		122
234.	351	041		21		122
235.	352	042		22		122
236.	353	043		23	Virgo	142
237.	354	044		24		142
238.	355	045		25		142
239.	356	046		26		142
240.	357	047		27		142
241.	360	050		28		142
242.	361	051		29		142
243.	362	060		30		142
244.	363	061		31		142
245.	364	101	Sept.	1		142
246.	365	102		2		142
247.	366	103		3		142
248.	367	104		4		142
249.	370	105		5		142
250.	371	106		6		142
251.	372	107		7		142
252.	373	110		8		142
253.	374	111		9		142

1702 READ ONLY MEMORY LOOK UP TABLE FOR CALENDAR BOARD

	address	least significant Byte data	comments		most significant Byte data
254.	375	120	Sept. 10		142
255.	376	121	11		142
256.	377	122	12		142
257.	000	123	13		142
258.	001	124	14		142
259.	002	125	15		142
260.	003	126	16		142
261.	004	127	17		142
262.	005	130	18		142
263.	006	131	19		142
264.	007	140	20		142
265.	010	141	21		142
266.	011	142	22		142
267.	012	143	23	Libra	162
268.	013	144	24		162
269.	014	145	25		162
270.	015	146	26		162
271.	016	147	27		162
272.	017	150	28		162
273.	020	151	29		162
274.	021	160	30		162
275.	022	001	Oct. 1		164
276.	023	002	2		164
277.	024	003	3		164
278.	025	004	4		164
279.	026	005	5		164
280.	027	006	6		164
281.	030	007	7		164
282.	031	010	8		164
283.	032	011	9		164
284.	033	020	10		164
285.	034	021	11		164
286.	035	022	12		164
287.	036	023	13		164
288.	037	024	14		164
289.	040	025	15		164

1702 READ ONLY MEMORY LOOK UP TABLE FOR CALENDAR BOARD

	address	least significant Byte data	comments		most significant Byte data
290.	041	026	Oct. 16		164
291.	042	027		17	164
292.	043	030		18	164
293.	044	031		19	164
294.	045	040		20	164
295.	046	041		21	164
296.	047	042		22	164
297.	050	043		23	164
298.	051	044		24 Scorpious	204
299.	052	045		25	204
300.	053	046		26	204
301.	054	047		27	204
302.	055	050		28	204
303.	056	051		29	204
304.	057	060		30	204
305.	060	061		31	204
306.	061	101	Nov. 1		204
307.	062	102		2	204
308.	063	103		3	204
309.	064	104		4	204
310.	065	105		5	204
311.	066	106		6	204
312.	067	107		7	204
313.	070	110		8	204
314.	071	111		9	204
315.	072	120		10	204
316.	073	121		11	204
317.	074	122		12	204
318.	075	123		13	204
319.	076	124		14	204
320.	077	125		15	204
321.	100	126		16	204
322.	101	127		17	204
323	102	130		18	204
324.	103	131		19	204
325.	104	140		20	204
326.	105	141		21	204
327.	106	142		22 Sagittar-	224
328.	107	143		23 ius	224

1702 READ ONLY LOOK UP TABLE FOR CALENDAR BOARD

	address	least significant Byte data	comments		most significant Byte data
329.	110	144	Nov. 24		224
330.	111	145	25		224
331.	112	146	26		224
332.	113	147	27		224
333.	114	150	28		224
334.	115	151	29		224
335.	116	160	30		224
336.	117	201	Dec. 1		224
337.	120	202	2		224
338.	121	203	3		224
339.	122	204	4		224
340.	123	205	5		224
341.	124	206	6		224
342.	125	207	7		224
343.	126	210	8		224
344.	127	211	9		224
345.	130	220	10		224
346.	131	221	11		224
347.	132	222	12		224
348.	133	223	13		224
349.	134	224	14		224
350.	135	225	15		224
351.	136	226	16		224
352.	137	227	17		224
353.	140	230	18		224
354.	141	231	19		224
355.	142	240	20		224
356.	143	241	21		224
357.	144	242	22	Capri-	244
358.	145	243	23	cornus	244
359.	146	244	24		244
360.	147	245	25		244
361.	150	246	26		244
362.	151	247	27		244
363.	152	250	28		244
364.	153	251	29		244
365.	154	260	30		244
366.	155	261	31		244
367.	156	262	Reset		254

APPENDIX C

Project Photographs

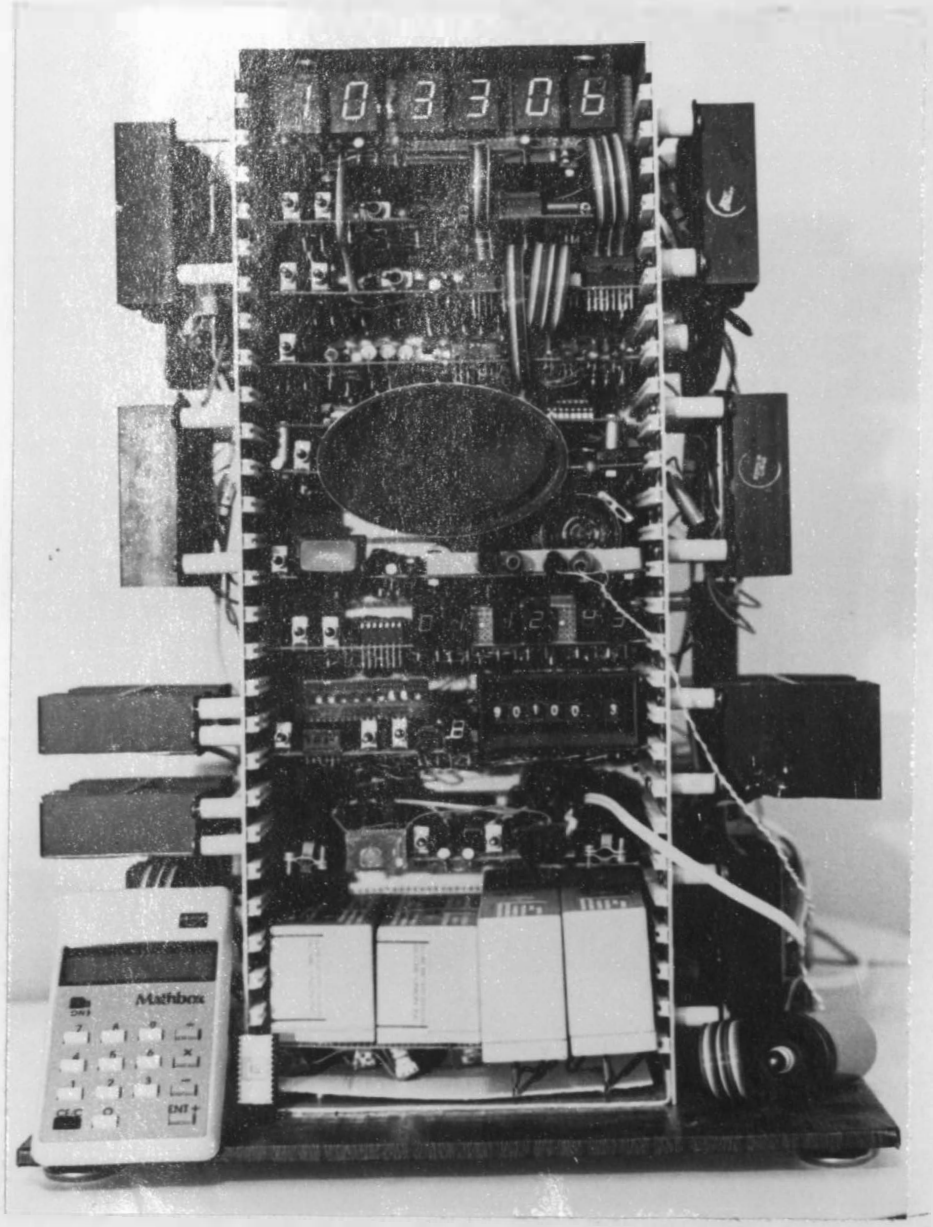


Figure C-1 Clock Front View

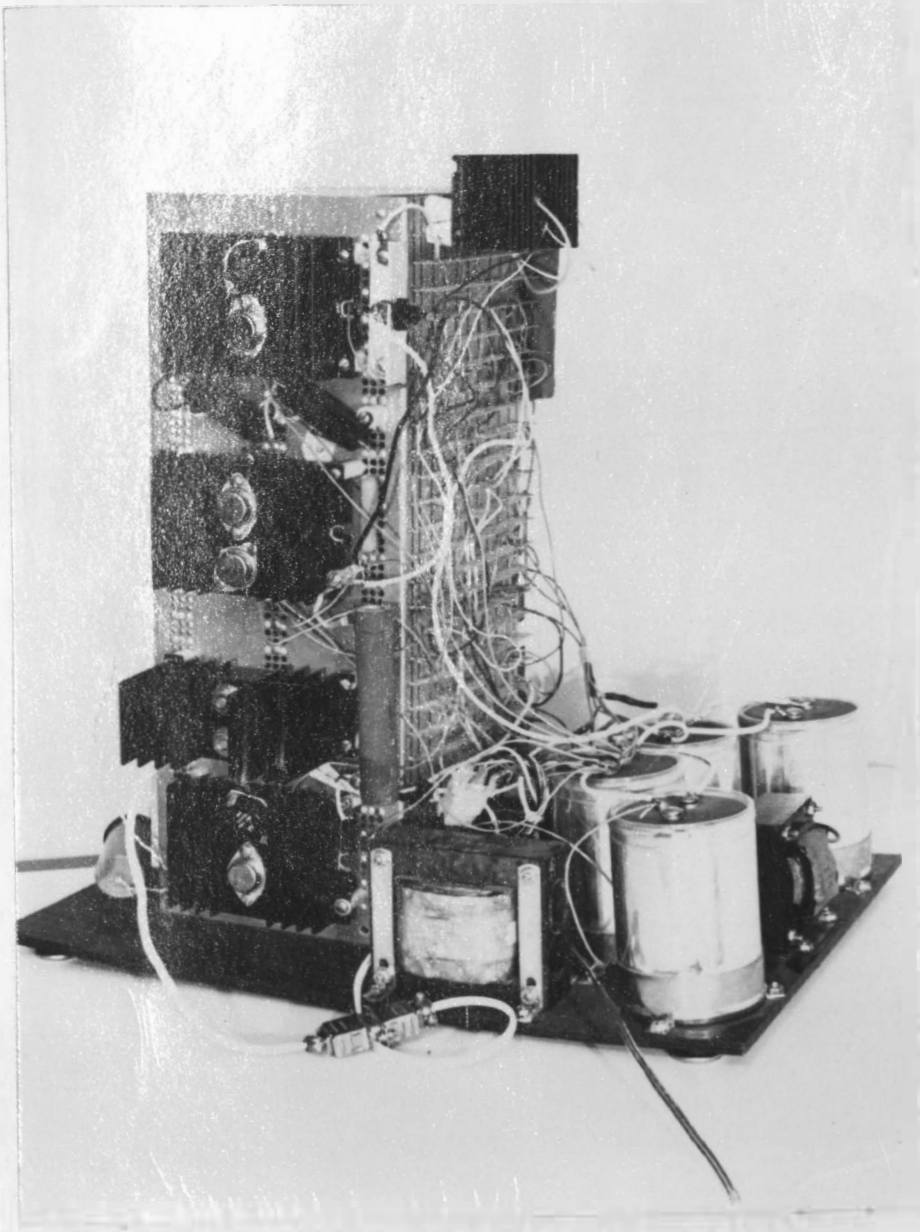


Figure C-2 Clock Side View

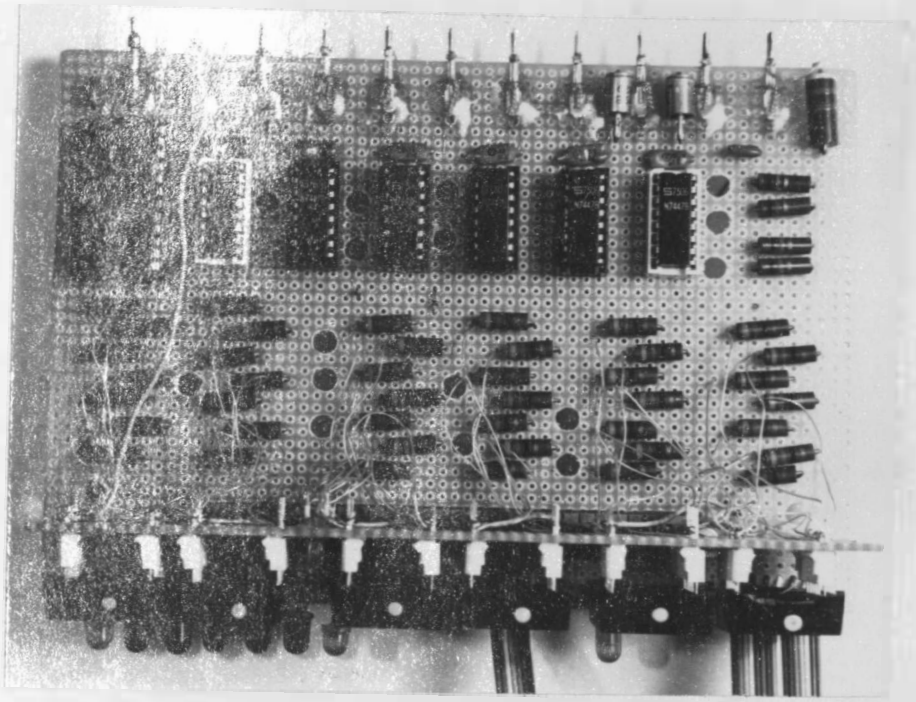


Figure C-3 Display Board

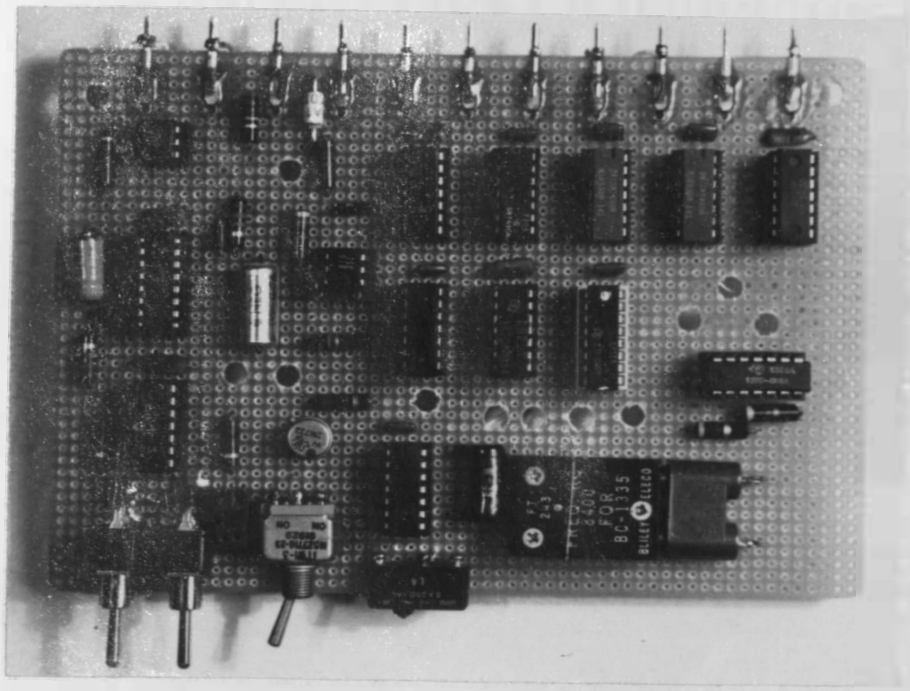


Figure C-4 Overflow Board

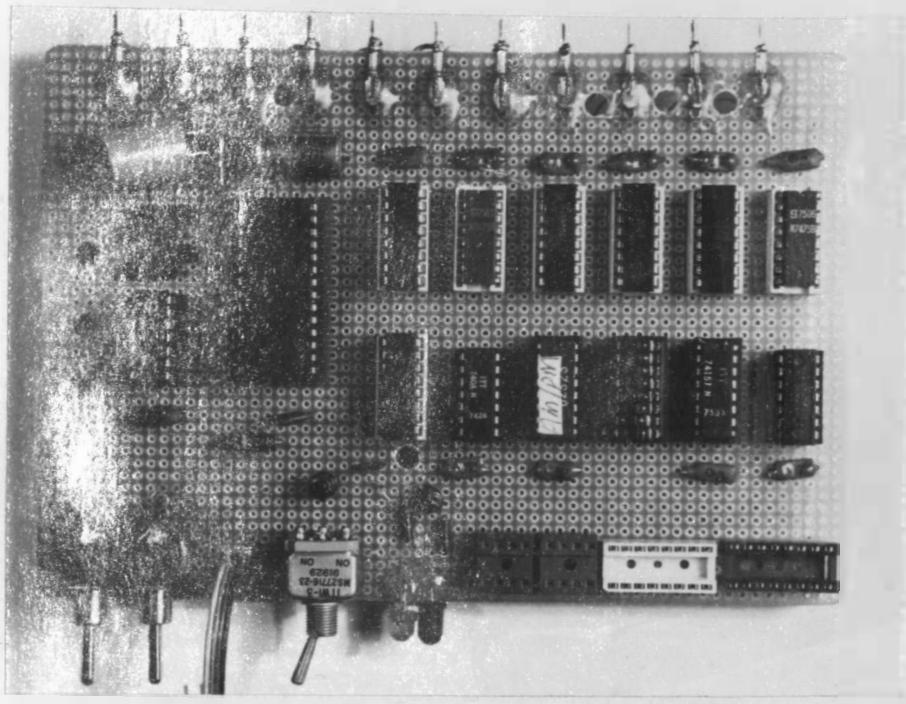


Figure C-5 Timing Board

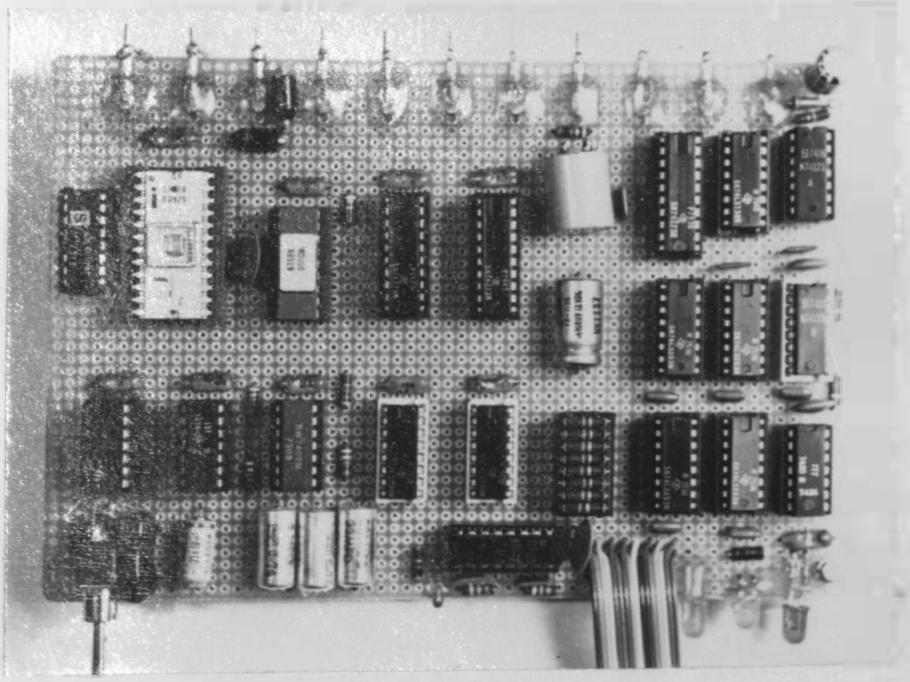


Figure C-6 Central Processing Unit Board

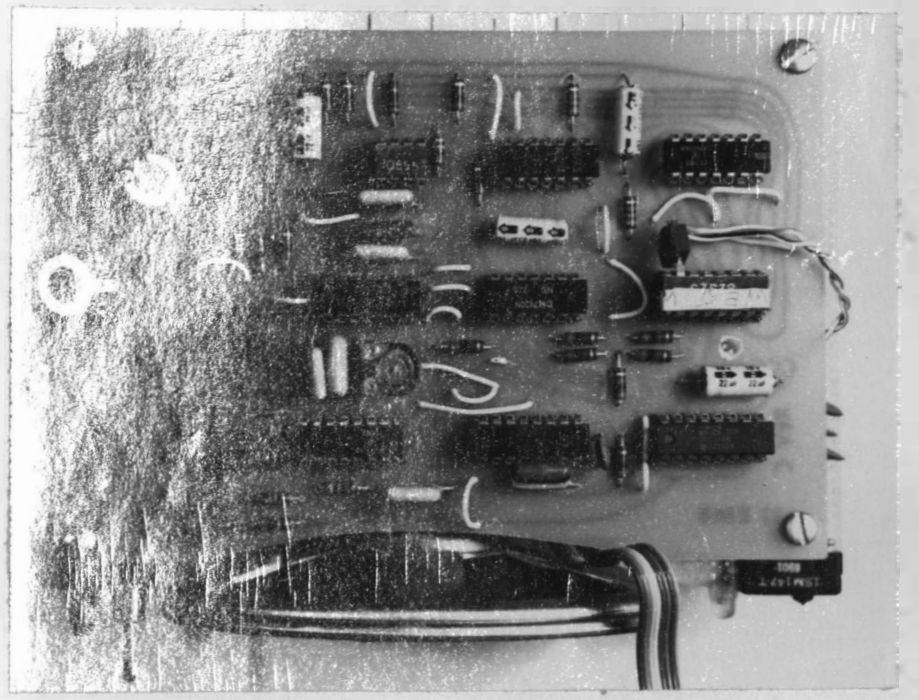


Figure C-7 Westminster Melody Board

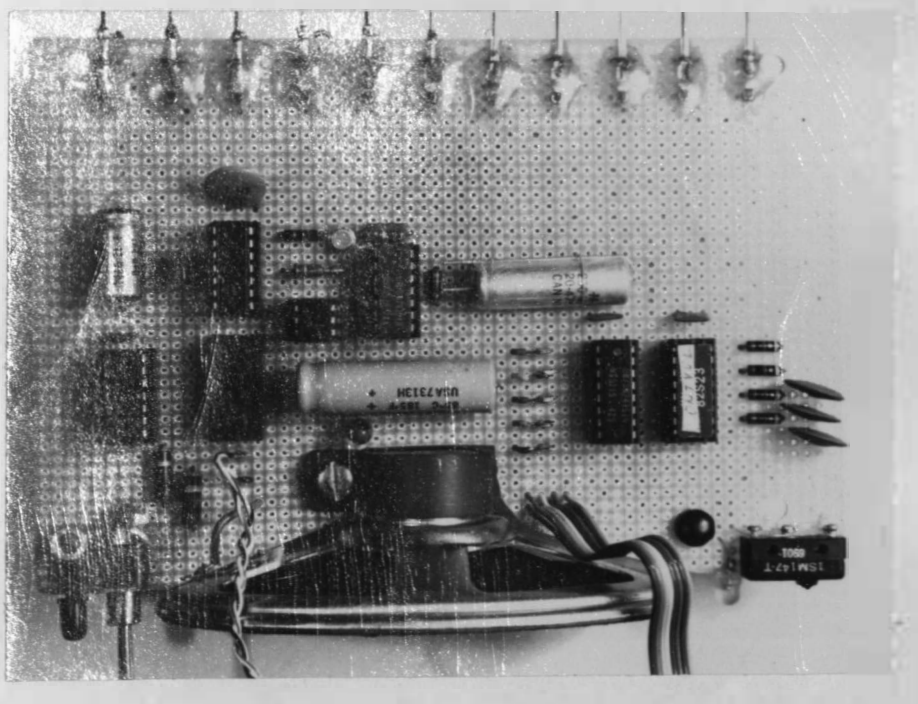


Figure C-8 Westminster Control Board

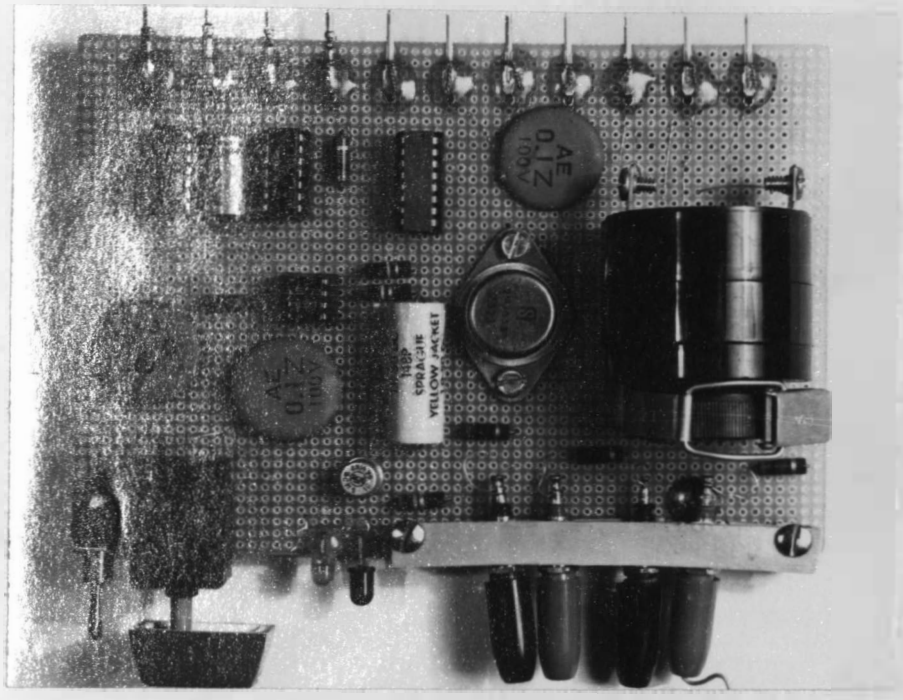


Figure C-9 Intrusion Board

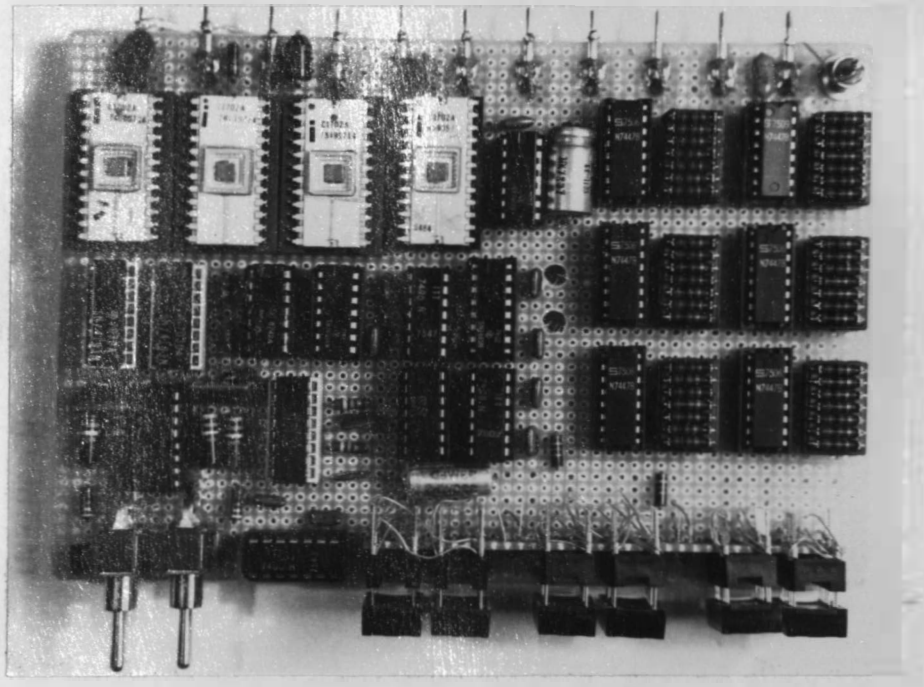


Figure C-10 Calendar Board

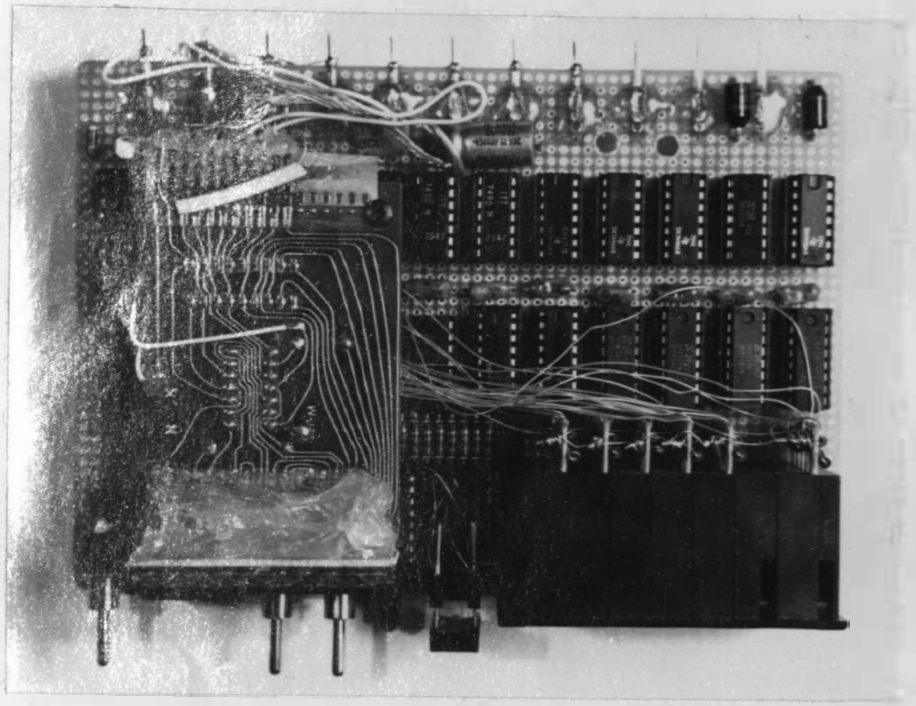


Figure C-11 Phase of the Moon Board (Assembled)

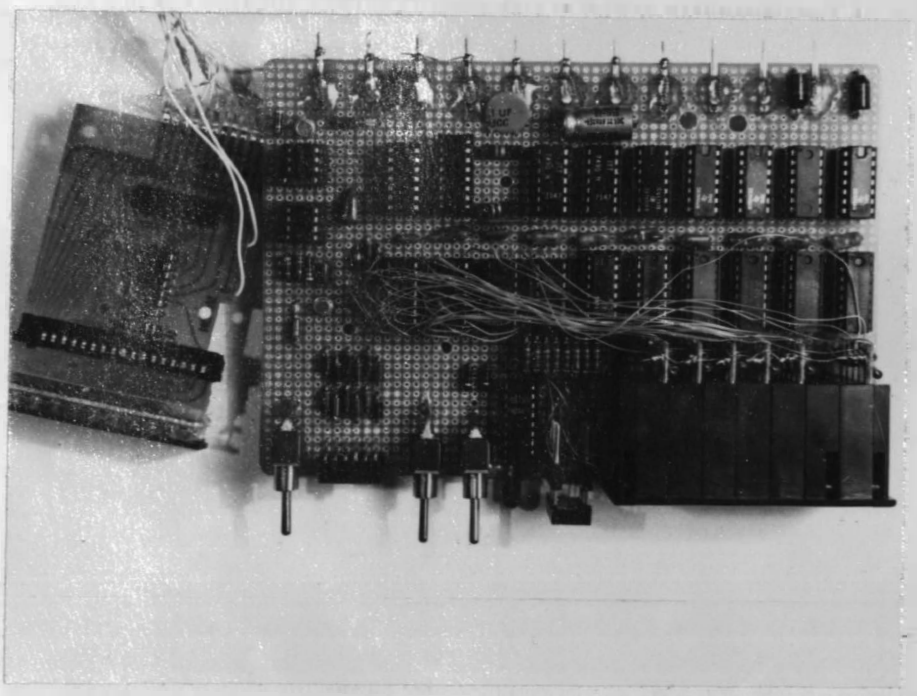


Figure C-12 Phase of the Moon Board (Disassembled)

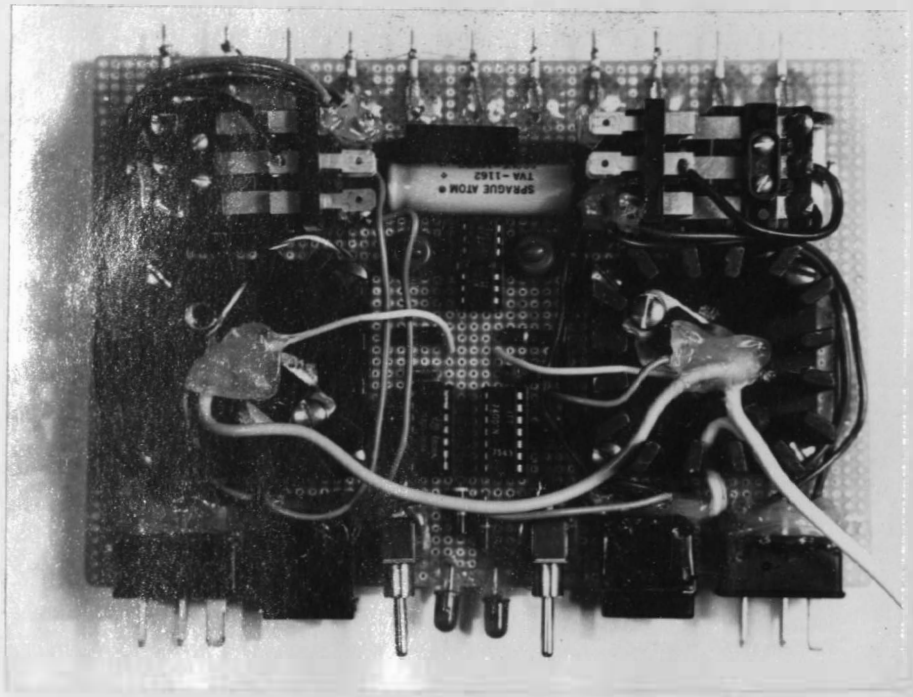


Figure C-13 Power Board

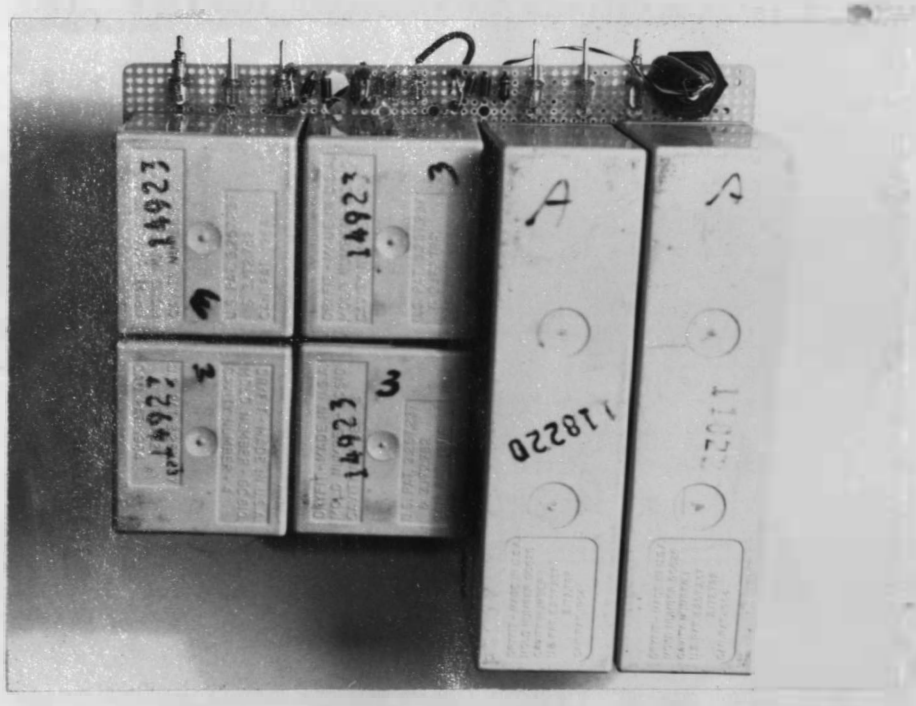


Figure C-14 Battery Board